Documentation of Questionnaire/Module 'FRS0405A' on 05-05-2004 at 15:54

FRS0405A

FAMILY RESOURCES SURVEY 2004/2005

COMPUTE ALWAYS:

```
Edit := No
```

COMPUTE ALWAYS:

```
NatCen := ONS
```

COMPUTE ALWAYS:

```
Test := No
```

COMPUTE ALWAYS:

```
VerCode := '054_1'
```

COMPUTE ALWAYS:

```
TestVer := '01'
```

COMPUTE ALWAYS:

```
SuppTxt := ('INTERVIEWER: PLEASE RECORD THE REASONS WHY YOU
SUPPRESSED ' + 'THIS WARNING, THEN PRESS <ALT> + S TO SAVE AND
CONTINUE.')
```

COMPUTE ALWAYS:

```
Pd97Txt := ('INTERVIEWER: PLEASE LEAVE A NOTE/REMARK GIVING
FULL DETAILS ' + 'THEN PRESS <ALT> + S TO SAVE AND CONTINUE.')
```

COMPUTE ALWAYS:

```
KeyTxt := ('INTERVIEWER: This is a 'Key Question': it is VERY '
+ 'IMPORTANT to get an answer here if possible. If you cannot '
+ 'do so (either now, or later) please make a Note about the '
+ 'circumstances.')
```

COMPUTE ALWAYS:

```
P := '£'
```

COMPUTE ALWAYS:

```
S := '·'
```

COMPUTE ALWAYS:

```
S4 := (S + S + S + S)
```

COMPUTE ALWAYS:

```
S10 := (S + S + S + S + S + S + S + S + S + S)
```

RECORD ALWAYS:

IVers

Version code of interview program, eg. I_048_1. I = Interview, 04 = month, 7 = year, 1 = release.

STRING[7]

RECORD ALWAYS:

EVers

Version code of edit program, eg. E_047_1. E = Edit, 04 = month, 7 = year, 1 = release.

STRING[7]

COMPUTE IF: Test = Yes
AND: NatCen = NI

```
IVers := ('NI_' + TestVer)
```

COMPUTE IF: Test = Yes
AND: NOT (NatCen = NI)

```
IVers := ('Test_' + TestVer)
```

COMPUTE IF: Edit = Yes

```
EVers := ('E_' + VerCode)
```

COMPUTE IF: NOT (Edit = Yes)

```
IVers := ('I_' + VerCode)
```

COMPUTE IF: Edit = Yes

```
Interviewer := 'EDITOR'
```

COMPUTE IF: Edit = Yes

```
EditVersion := ('Edit version :@|@|' + EVers + '
')
```

COMPUTE IF: NOT (Edit = Yes)

```
Interviewer := 'INTERVIEWER'
```

COMPUTE IF: NOT (Edit = Yes)

```
EditVersion := '
'
```

COMPUTE ALWAYS:

```
Days[1] := 'Sunday'
```

COMPUTE ALWAYS:

```
Days[2] := 'Monday'
```

COMPUTE ALWAYS:

```
Days[3] := 'Tuesday'
```

COMPUTE ALWAYS:

```
Days[4] := 'Wednesday'
```

COMPUTE ALWAYS:

```
Days[5] := 'Thursday'
```

COMPUTE ALWAYS:

```
Days[6] := 'Friday'
```

COMPUTE ALWAYS:

```
Days[7] := 'Saturday'
```

COMPUTE ALWAYS:

```
Months[1] := 'January'
```

COMPUTE ALWAYS:

```
Months[2] := 'February'
```

COMPUTE ALWAYS:

```
Months[3] := 'March'
```

COMPUTE ALWAYS:

```
Months[4] := 'April'
```

COMPUTE ALWAYS:

```
Months[5] := 'May'
```

COMPUTE ALWAYS:

```
Months[6] := 'June'
```

COMPUTE ALWAYS:

```
Months[7] := 'July'
```

COMPUTE ALWAYS:

```
Months[8] := 'August'
```

COMPUTE ALWAYS:

```
Months[9] := 'September'
```

COMPUTE ALWAYS:

```
Months[10] := 'October'
```

COMPUTE ALWAYS:

```
Months[11] := 'November'
```

COMPUTE ALWAYS:

```
Months[12] := 'December'
```

COMPUTE ALWAYS:

```
AssDo := No
```

COMPUTE ALWAYS:

```
BookDo := No
```

COMPUTE ALWAYS:

```
NCDVLP := No
```

COMPUTE ALWAYS:

```
NCDVIB := 0
```

COMPUTE ALWAYS:

```
NCDVOB := 0
```

COMPUTE ALWAYS:

```
NCDVDC := No
```

COMPUTE ALWAYS:

```
NCDVTC := No
```

COMPUTE ALWAYS:

```
NCDVCP := 0
```

COMPUTE ALWAYS:

```
NCDVAW := No
```

COMPUTE ALWAYS:

```
NCDVRT := No
```

COMPUTE ALWAYS:

```
NCDVAA := No
```

FRS0405A.QSerial

Serial number

ASK ALWAYS:

Area

AREA NUMBER.

JUST PRESS <Enter>.

1..99997

ASK ALWAYS:

Address

ADDRESS NUMBER.

JUST PRESS <Enter>.

1..97

ASK ALWAYS:

Hhold

HOUSEHOLD NUMBER.

JUST PRESS <Enter>.

1..3

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

COMPUTE IF: *DArea > 0*

Area := DArea

COMPUTE IF: *DAddress > 0*

Address := DAddress

COMPUTE IF: *DHhold > 0*

```
Hhold := DHhold
```

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

CHECK ALWAYS:
    RESERVECHECK

    RESERVECHECK

CHECK ALWAYS:
    RESERVECHECK

    RESERVECHECK

CHECK ALWAYS:
    RESERVECHECK

    RESERVECHECK

CHECK ALWAYS:
    RESERVECHECK

    RESERVECHECK

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: QSerial.Area < 10

StrArea := ('0000' + STR(QSerial.Area,1))
```

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: QSerial.Area < 100

StrArea := ('000' + STR(QSerial.Area,2))
```

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: QSerial.Area < 1000

StrArea := ('00' + STR(QSerial.Area,3))
```

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: QSerial.Area < 10000

StrArea := ('0' + STR(QSerial.Area,4))
```

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: NOT (QSerial.Area < 10000)

StrArea := STR(QSerial.Area,5)
```

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: QSerial.Address IN [1 .. 9]

StrAddr := ('0' + STR(QSerial.Address,1))
```

```
COMPUTE IF: (QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)
     AND: NOT (QSerial.Address IN [1 .. 9])

StrAddr := STR(QSerial.Address,2)
```

COMPUTE IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

```
StrSerial := (StrArea + StrAddr)
```

COMPUTE IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

```
Serial := VAL(StrSerial)
```

FRS0405A.QDataBag

Sample information

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

Serial

Serial number excluding household number.

1..9999997

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

Hhold

Household number

1..3

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

IntNo

Interviewer Number

0..9999

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

SurvId

3-letter acronym for survey.

STRING[3]

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

SampYear

Year Code

1998..9997

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

SampMnth

Sample month.

1..12

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

SampQtr

Sample quarter

1..4

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

Attempt

Issue number.

1..7

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

SSTRTReg

Stratifying region: Survey specific.

1..97

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

StaRegGB

Statistical region in GB.

1..12

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

GovRegGB

Government office region in GB.

1..12

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

Country

1..7

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

ACORN

0..97

ASK IF: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

SLA

Local Authority Code.

STRING[4]

Ask if: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

LAC

Local Authority Code. GOV version

0..997

Ask if: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

NICoun

Northern Ireland District Council Codes

1..97

Ask if: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

NIRate

Northern Ireland District Council Rates

-99.9999..999.9999

Ask if: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

NINRV

Net rateable value of property (in N. Ireland)

1..9997

Ask if: *(QSerial.Area = RESPONSE) AND (QSerial.Address = RESPONSE)*

Spare

Spare code frame

1..7

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
ASK IF: Test = Yes
```

Country

QUESTION ONLY VISIBLE FOR TESTING PURPOSES:
Which Country?

(1) England
(2) Wales
(3) Scotland
(4) Northern Ireland

```
ASK IF: Test = Yes
     AND: Country = NIreland
```

NICoun

QUESTION ONLY VISIBLE FOR TESTING PURPOSES:

Enter Northern Ireland District Council code (01-26).

(1) Antrim
(2) Ards
(3) Armagh
(4) Ballymena
(5) Ballymoney
(6) Banbridge
(7) Belfast
(8) Carrickfergus
(9) Castlereagh
(10) Coleraine
(11) Cookstown
(12) Craigavon
(13) Downpatrick
(14) Dungannon
(15) Fermanagh
(16) Larne
(17) Limavady
(18) Lisburn
(19) Derry
(20) Magherafelt
(21) Moyle
(22) Newry
(23) Newtownabbey
(24) North Down
(25) Omagh
(26) Strabane
(27) <NOT USED>
(28) <NOT USED>
(29) <NOT USED>
(30) <NOT USED>

```
COMPUTE IF: Test = Yes
     AND: Country = Scotland
```

NIreland := No

COMPUTE IF: Test = Yes
AND: Country = Scotland

Scotland := Yes

COMPUTE IF: Test = Yes
AND: Country = Scotland

Wales := No

COMPUTE IF: Test = Yes
AND: Country = Wales

NIreland := No

COMPUTE IF: Test = Yes
AND: Country = Wales

Scotland := No

COMPUTE IF: Test = Yes
AND: Country = Wales

Wales := Yes

COMPUTE IF: Test = Yes
AND: Country = NIreland

NIreland := Yes

COMPUTE IF: Test = Yes
AND: Country = NIreland

Scotland := No

COMPUTE IF: Test = Yes
AND: Country = NIreland

Wales := Yes

COMPUTE IF: Test = Yes
AND: Country = NIreland

NatCen := NI

COMPUTE IF: Test = Yes
AND: Country = England

NIreland := No

COMPUTE IF: Test = Yes
AND: Country = England

Scotland := No

COMPUTE IF: Test = Yes
AND: Country = England

Wales := No

COMPUTE IF: QDataBag.SSTRTReg IN [22 .. 27]

NIreland := No

COMPUTE IF: QDataBag.SSTRTReg IN [22 .. 27]

Scotland := Yes

COMPUTE IF: QDataBag.SSTRTReg IN [22 .. 27]

Wales := No

COMPUTE IF: QDataBag.SSTRTReg IN [20 .. 21]

NIreland := No

COMPUTE IF: QDataBag.SSTRTReg IN [20 .. 21]

Scotland := No

COMPUTE IF: QDataBag.SSTRTReg IN [20 .. 21]

Wales := Yes

COMPUTE IF: QDataBag.SSTRTReg IN [30]

NIreland := Yes

COMPUTE IF: QDataBag.SSTRTReg IN [30]

Scotland := No

COMPUTE IF: QDataBag.SSTRTReg IN [30]

Wales := No

COMPUTE IF: NOT (QDataBag.SSTRTReg IN [30])

NIreland := No

COMPUTE IF: NOT (QDataBag.SSTRTReg IN [30])

Scotland := No

COMPUTE IF: NOT (QDataBag.SSTRTReg IN [30])

Wales := No

COMPUTE IF: Test = Yes
AND: NICoun = RESPONSE

NIDCoun := ORD(NICoun)

COMPUTE IF: QDataBag.NICoun IN [1 .. 97.00000000000001]

NIDCoun := QDataBag.NICoun

COMPUTE IF: QDataBag.NIRate IN [0.01 .. 9997]

NIRate := QDataBag.NIRate

COMPUTE IF: NatCen = NI

SharOwn := 'co-ownership'

COMPUTE IF: NatCen = NI

SOwners := 'CO-OWNERS'

COMPUTE IF: NatCen = NI

LANIHE := 'NIHE'

COMPUTE IF: NatCen = NI

Council1 := 'Northern Ireland Housing Executive'

COMPUTE IF: NatCen = NI

Council2 := 'Northern Ireland Housing Executive'

COMPUTE IF: NatCen = NI

GOVSSA := 'Social Security Agency'

COMPUTE IF: NatCen = NI

GOV1 := 'SSA'

COMPUTE IF: NatCen = NI

GOV2 := 'Social Security Agency'

COMPUTE IF: NatCen = NI

JobCen := 'a Social Security Office'

COMPUTE IF: NatCen = NI

RentReb1 := 'rent and/or rates rebate'

COMPUTE IF: NatCen = NI

RentReb2 := 'rent/rates rebate'

COMPUTE IF: NatCen = NI

LAuths := 'Social Services'

COMPUTE IF: NatCen = NI

LAuth1 := 'Social Services'

COMPUTE IF: NatCen = NI

LAuth2 := 'Social Services'

COMPUTE IF: NatCen = NI

IncROI1 := '
INCLUDE ACCOUNTS HELD IN THE REPUBLIC OF IRELAND.'

COMPUTE IF: NatCen = NI

```
IncROI2 := ('IF ACCOUNT HELD IN THE REPUBLIC OF IRELAND OPEN
NOTE ' + 'TO STATE IF AMOUNT RECORDED IN PUNTS OR EUROS.')
```

COMPUTE IF: NatCen = NI

```
IncROI3 := ('INCLUDE HOLDINGS AND/OR ACCOUNTS HELD IN THE
REPUBLIC ' + 'OF IRELAND AND OPEN NOTE TO STATE IF AMOUNT
RECORDED IN ' + 'PUNTS OR EUROS.')
```

COMPUTE IF: NatCen = NI

```
Dept := 'Department for Social Development'
```

COMPUTE IF: NatCen = NI

```
Mid_Pri := '< NOT USED >'
```

COMPUTE IF: NatCen = NI

```
Mid_Sec := '< NOT USED >'
```

COMPUTE IF: NatCen = NI

```
Grammar := '/Grammar'
```

COMPUTE IF: NatCen = NI

```
State_run := ''
```

COMPUTE IF: NatCen = NI

```
assisted := ''
```

COMPUTE IF: NatCen = NI

```
Inland_Revenue := 'Inland Revenue'
```

COMPUTE IF: NOT (NatCen = NI)

```
SharOwn := 'shared ownership'
```

COMPUTE IF: NOT (NatCen = NI)

```
SOwners := 'SHARED OWNERS'
```

COMPUTE IF: NOT (NatCen = NI)

```
LANIHE := 'local authority'
```

COMPUTE IF: NOT (NatCen = NI)

```
Council1 := ('The local authority/council/New Town ' +
'development/Scottish Homes')
```

COMPUTE IF: NOT (NatCen = NI)

```
Council2 := 'Local Authority or Council (incl. GLC)'
```

COMPUTE IF: NOT (NatCen = NI)

GOVSSA := 'DWP (formerly DSS)'

COMPUTE IF: NOT (NatCen = NI)

GOV1 := 'DWP'

COMPUTE IF: NOT (NatCen = NI)

GOV2 := 'DWP (formerly DSS)'

COMPUTE IF: NOT (NatCen = NI)

JobCen := 'an Employment Service local office or Jobcentre'

COMPUTE IF: NOT (NatCen = NI)

RentReb1 := 'rent rebate'

COMPUTE IF: NOT (NatCen = NI)

RentReb2 := 'rent rebate'

COMPUTE IF: NOT (NatCen = NI)

LAuths := 'Local Authorities'

COMPUTE IF: NOT (NatCen = NI)

LAuth1 := 'Local Authority'

COMPUTE IF: NOT (NatCen = NI)

LAuth2 := 'a Local Authority'

COMPUTE IF: NOT (NatCen = NI)

IncROI1 := ''

COMPUTE IF: NOT (NatCen = NI)

IncROI2 := ''

COMPUTE IF: NOT (NatCen = NI)

IncROI3 := ''

COMPUTE IF: NOT (NatCen = NI)

Dept := 'Department for Work and Pensions'

COMPUTE IF: NOT (NatCen = NI)

Mid_Pri := 'Middle-deemed primary school (state run or assisted)'

COMPUTE IF: NOT (NatCen = NI)

Mid_Sec := 'Middle-deemed secondary school (state run or assisted)'

COMPUTE IF: NOT (NatCen = NI)

Grammar := ''

COMPUTE IF: NOT (NatCen = NI)

State_run := 'State run'

COMPUTE IF: NOT (NatCen = NI)

assisted := '(State run or assisted)'

COMPUTE IF: NOT (NatCen = NI)

Inland_Revenue := 'Inland Revenue (or formerly the DSS)'

ASK ALWAYS:

First

^INTERVIEWER: FOR YOUR INFORMATION... You are in the Household Schedule for
Area No:@|@|^StrArea
Address No:@|@|^StrAddr
Household No:@|^QSerial.Hhold

- TO GO DIRECTLY TO 'ADMIN', PRESS <Ctrl + Enter>.
- TO CONTINUE WITH INTERVIEW PRESS '1' AND <Enter>.

^S Questionnaire version :@|^IVers
^S ^EditVersion

(1) Continue

ASK IF: NatCen = Yes

AdrCheck

REFER TO ADDRESS LABEL:
CHECK THAT LABEL GIVES RESPONDENT'S FULL CURRENT ADDRESS.

IF NOT, AMEND ADDRESS AND CODE 'Address changed'.

(1) Address confirmed
(2) Address changed

ASK IF: Test = Yes

ThisYear

Enter FRS Survey Year (eg. 2003 = April 2003 - March 2004).

2003..2010

Ask if: *Test = Yes*

ThisMnth

Enter survey month.

1..12

Compute if: *(QDataBag.SampYear = RESPONSE) AND (QDataBag.SampMnth = RESPONSE)*

ThisYear := QDataBag.SampYear

Compute if: *(QDataBag.SampYear = RESPONSE) AND (QDataBag.SampMnth = RESPONSE)*

ThisMnth := QDataBag.SampMnth

Compute if: *ThisMnth IN [1 .. 12]*

MnthOK2 := ThisMnth

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 11*

MnthOK1 := 10

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 11*

MnthOK3 := 12

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 11*

MnthOK4 := 1

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 12*

MnthOK1 := 11

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 12*

MnthOK3 := 1

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 12*

MnthOK4 := 2

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 1*

MnthOK1 := 12

Compute if: *ThisMnth IN [1 .. 12]*
And: *ThisMnth = 1*

MnthOK3 := 2

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: ThisMnth = 1
```

MnthOK4 := 3

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: ThisMnth = 4
```

MnthOK1 := 4

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: ThisMnth = 4
```

MnthOK3 := 5

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: ThisMnth = 4
```

MnthOK4 := 6

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: NOT (ThisMnth = 4)
```

MnthOK1 := (ThisMnth - 1)

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: NOT (ThisMnth = 4)
```

MnthOK3 := (ThisMnth + 1)

```
COMPUTE IF: ThisMnth IN [1 .. 12]
      AND: NOT (ThisMnth = 4)
```

MnthOK4 := (ThisMnth + 2)

FRS0405A.QSignIn

Record always:

StartDat

THE DATE ON WHICH THE INTERVIEW WITH THIS HOUSEHOLD WAS STARTED.

DATE

Record always:

DateOK

INTERVIEWER: TODAY'S DATE ACCORDING TO THE LAPTOP IS ^AStartD.

IS THIS THE CORRECT DATE?

(1) Yes
(2) No

Compute if: *StartDat = EMPTY OR (Test = Yes)*

AStartD := SYSDATE

Ask if: *StartDat = EMPTY OR (Test = Yes)*

DateOK

INTERVIEWER: TODAY'S DATE ACCORDING TO THE LAPTOP IS ^AStartD.

IS THIS THE CORRECT DATE?

(1) Yes
(2) No

Compute if: *StartDat = EMPTY OR (Test = Yes)*
And: *DateOK = Yes*

StartDat := AStartD

Ask if: *StartDat = EMPTY OR (Test = Yes)*
And: *DateOK = No*

BStartD

ENTER THE DATE ON WHICH THE INTERVIEW WITH THIS HOUSEHOLD WAS STARTED.

DATE

COMPUTE IF: *StartDat = EMPTY OR (Test = Yes)*
AND: *DateOK = No*
AND: *(((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND (((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4))) OR (ThisYear <> RESPONSE)) OR (ThisMnth <> RESPONSE)*

StartDat := BStartD

CHECK IF: *StartDat = EMPTY OR (Test = Yes)*
AND: *DateOK = No*
AND: *NOT ((((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND (((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4))) OR (ThisYear <> RESPONSE)) OR (ThisMnth <> RESPONSE)*
((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND INVOLVING(BStartD)

The year is wrong for the current FRS survey (^ThisYear)!
Please re-enter the date from the beginning.

CHECK IF: *StartDat = EMPTY OR (Test = Yes)*
AND: *DateOK = No*
AND: *NOT ((((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND (((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4))) OR (ThisYear <> RESPONSE)) OR (ThisMnth <> RESPONSE)*
((((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4)) AND INVOLVING(BStartD)

The month is wrong for the current FRS survey (^Months[ThisMnth])!
Please re-enter the date from the beginning.

DISPLAY ALWAYS:

StartDat

THE DATE ON WHICH THE INTERVIEW WITH THIS HOUSEHOLD WAS STARTED.

DATE

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

RECORD ALWAYS:

IntSTime

Interview start time

TIME

COMPUTE IF: *IntSTime = EMPTY AND StartDat <> EMPTY*

IntSTime := STARTTIME

ASK IF: *(Edit = Yes) AND (NatCen <> Yes)*

Editor

EDITOR at HQ: Enter your identification number.

1..97

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

CHECK ALWAYS:
```
     RESERVECHECK
```

RESERVECHECK

CHECK ALWAYS:
```
     RESERVECHECK
```

RESERVECHECK

CHECK ALWAYS:
```
     RESERVECHECK
```

RESERVECHECK

CHECK ALWAYS:
```
     RESERVECHECK
```

RESERVECHECK

COMPUTE IF: QDataBag.SampMnth IN [1, 2, 3]

```
CheckYear := 2005
```

COMPUTE IF: NOT (QDataBag.SampMnth IN [1, 2, 3])

```
CheckYear := 2004
```

COMPUTE ALWAYS:

```
FWDate := TODATE(CheckYear,QDataBag.SampMnth,1)
```

WARN IF: QDataBag.SampYear <> 0
```
     (QSignIn.StartDat.YEAR = FWDate.YEAR) OR ((QSignIn.StartDat.YEAR =
     (FWDate.YEAR + 1)) AND (QSignIn.StartDat <= (FWDate + (0,3,0))))
```

You have accidentally entered the wrong year. It doesn't agree with the fieldwork period. Please check and amend.

WARN IF: QDataBag.SampMnth <> 0
```
     (QSignIn.StartDat >= FWDate) OR ((FWDate.MONTH = 12) AND
     (QSignIn.StartDat.MONTH = 11))
```

You have accidentally entered the wrong month and/or year. It doesn't agree with the fieldwork period. Please check and amend.

WARN IF: QDataBag.SampMnth <> 0
```
     QSignIn.StartDat <= (FWDate + (0,3,0))
```

The month of this date is more than 3 months after the fieldwork period, please check and amend.

COMPUTE ALWAYS:

```
DLYear := (QSignIn.StartDat + (-1,0,0))
```

FRS0405A.QNames

Names of household members

ASK ALWAYS:

WhoHere

Who normally lives at this address?

(1) Press <Enter> to continue.

FRS0405A.QNames.M[]

```
ASK IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
```

Name

ENTER AN IDENTIFIER FOR THIS HOUSEHOLD MEMBER

IT DOESN'T HAVE TO BE A NAME - JUST SOMETHING THAT UNIQUELY IDENTIFIES THIS PERSON WITHIN THE HOUSEHOLD SO YOU CAN REFER TO THEM LATER IN THE INTERVIEW.

STRING[15]

```
ASK IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: PPers < 14
```

More

Is there anyone else in this household?

(1) Yes
(2) No

FRS0405A.QNames (continued)

Names of household members

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: M[Pers].More = No

HSize := Pers
```

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

RECORD ALWAYS:

HHSize

Household size including any x-ed out

0..14

COMPUTE IF: *(QNames.HSize > 0) OR (Edit = No)*

HHSize := QNames.HSize

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*

DMName[Loop1] := QNames.M[Loop1].Name

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

FRS0405A.HHG

Data on household members

COMPUTE IF: HHSize > 0

FHHSize := PHHSize

COMPUTE IF: HHSize > 0
AND: NatCen = Yes

Out772 := '772'

COMPUTE IF: HHSize > 0
AND: NOT (NatCen = Yes)

Out772 := '77'

COMPUTE IF: HHSize > 0
AND: In loop FOR P1 := 1 TO FHHSize

P[P1].Person := P1

COMPUTE IF: HHSize > 0
AND: In loop FOR P1 := 1 TO FHHSize

P[P1].Name := QNames.M[].Name[P1]

FRS0405A.HHG.P[]

Record if: *HHSize > 0*
    And: *In loop FOR P1 := 1 TO FHHSize*

BenUnit

HHG
Benefit Unit number.

0..7

Record if: *HHSize > 0*
    And: *In loop FOR P1 := 1 TO FHHSize*

Person

HHG
Person number in Household Grid.

0..14

Record if: *HHSize > 0*
    And: *In loop FOR P1 := 1 TO FHHSize*

Name

HHG
First name.

STRING[15]

Compute if: *HHSize > 0*
    And: *In loop FOR P1 := 1 TO FHHSize*

LName := Name

Compute if: *HHSize > 0*
    And: *In loop FOR P1 := 1 TO FHHSize*

UName := UPCASE(Name)

Ask if: *HHSize > 0*
    And: *In loop FOR P1 := 1 TO FHHSize*

Sex

HHG

INTERVIEWER: CODE ^UName'S SEX.

(1) Male
(2) Female

DISPLAY IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*

Name

HHG
First name.

STRING[15]

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    AND: *Sex = Male*

heshe := 'he'

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    AND: *Sex = Male*

hisher := 'his'

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    AND: *NOT (Sex = Male)*

heshe := 'she'

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    AND: *NOT (Sex = Male)*

hisher := 'her'

ASK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*

AgeOf

HHG

What was ^LName's age last birthday?

IF AGE NOT GIVEN, PROBE FOR AN ESTIMATE.
FOR LATER ROUTING, YOU MUST KNOW WHETHER:
A)^S^S^S MEN ARE AGED 16-64 OR 65+
B)^S^S^S WOMEN ARE AGED 16-59 OR 60+

0..120

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*

DVAge := AgeOf

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: AgeOf IN [16 .. 120]
```

MS

HHG

The aim is to obtain legal marital status, irrespective of any de facto arrangement. The only qualification to this aim is that you should not probe the answer 'separated'. Should a respondent query the term, explain that it covers any person whose spouse is living elsewhere because of estrangement (whether the separation is legal or not). Ignore temporary absences, eg. on oil rig.
A person whose spouse has been working away from home for over 6 months, eg. on a contract overseas or in the armed forces, should still be coded as married and living with husband/wife if the separation is not permanent.

(1) ^S^S^S single, that is, never married,
(2) ^S^S^S married and living with husband/wife, ^S^S^S HELP <F9>
(3) ...married and separated from husband/wife,
(4) ...divorced,
(5) ...or widowed?

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: AgeOf IN [16 .. 120]
     AND: (FHHSize > 1) AND (MS IN [Single, Sep, Divorce, Widowed])
```

CupChk

HHG

May I just check, are you / is ^LName living with someone in this household as a couple? ASK OR RECORD.

INTERVIEWER INSTRUCTION:
Only respondents who are living with their partner should be coded as living together as a couple.
You may code No without asking the question ONLY if all members of the household are too closely related for any to be living together in a de facto marital relationship.

(1) Yes
(2) No
(3) SPONTANEOUS ONLY - same sex couple

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
```

W1

HHG

What was ^LName's age when widowed?

0..120

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
     AND: W1 = RESPONSE
     W1 >= 16
```

Are you sure? It is not usual to be married before the age of 16 unless you were married outside the UK.

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
     AND: W1 = RESPONSE
     W1 <= AgeOf
```

You've coded that ^LName is ^AgeOf years old, but was widowed at the age of ^W1. Please amend the one or the other.

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
```

W2

HHG

Did ^LName have any children aged under 16 when widowed?

(1) Yes
(2) No

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

SonDaughter := 'son'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

BrotherSister := 'brother'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

FatherMother := 'father'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT (Sex = Male)
```

SonDaughter := 'daughter'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT (Sex = Male)
```

BrotherSister := 'sister'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT (Sex = Male)
```

FatherMother := 'mother'

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

FRS0405A.HHG.P[].QRel[]

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: RPers < PPers
```

R

HHG

ASK OR CODE ^PName's RELATIONSHIP TO ^RName.

(1)	spouse,
(2)	cohabitee,
(3)	^SonDaughter (incl. adopted) (/legal dependant),
(4)	step-^SonDaughter,
(5)	foster child,
(6)	^SonDaughter-in-law,
(7)	^FatherMother (or guardian),
(8)	step-^FatherMother,
(9)	foster parent,
(10)	^FatherMother-in-law,
(11)	^BrotherSister (incl. adopted),
(12)	step-^BrotherSister,
(13)	foster ^BrotherSister,
(14)	^BrotherSister-in-law,
(15)	grand-^SonDaughter,
(16)	grand-^FatherMother,
(17)	other relative,
(18)	or other non-relative?
(97)	(THIS CODE NOT USED)

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: RPers < PPers
     R <> Self
```

Code 97 is not valid for this question.

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: NOT (RPers < PPers)
```

R

HHG

ASK OR CODE ^PName's RELATIONSHIP TO ^RName.

(1)	spouse,
(2)	cohabitee,
(3)	^SonDaughter (incl. adopted) (/legal dependant),
(4)	step-^SonDaughter,
(5)	foster child,
(6)	^SonDaughter-in-law,
(7)	^FatherMother (or guardian),
(8)	step-^FatherMother,
(9)	foster parent,
(10)	^FatherMother-in-law,
(11)	^BrotherSister (incl. adopted),
(12)	step-^BrotherSister,
(13)	foster ^BrotherSister,
(14)	^BrotherSister-in-law,
(15)	grand-^SonDaughter,
(16)	grand-^FatherMother,
(17)	other relative,
(18)	or other non-relative?
(97)	(THIS CODE NOT USED)

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: NOT (RPers < PPers)
     AND: RPers = PPers
```

R := Self

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     NOT(IN(R,[???,???,???]))
```

Warning: This code must only be used if the foster child is covered by a Local Authority Allowance. Please check that this is the case.

FRS0405A.HHG.P[] (continued)

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

Spouses

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

NumParn

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

NumPart

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

NumCohab

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

Parent1

HHG
Person number of parent 1

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

Parent2

HHG
Person number of parent 2

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

Hholder

HHG
Is this person coded at QHholder.Hhldr?

(1) Yes
(2) No

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: AgeOf IN [4 .. 74]
```

FTEd

HHG

Is ^LName currently in full-time education?

INTERVIEWER: INCLUDE CORRESPONDENCE COURSES AND OPEN LEARNING AS WELL AS OTHER FORMS OF FULL-TIME COURSES.

(1) Yes
(2) No

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: AgeOf IN [19 .. 23]
```

StillEduc := (' - or is ' + heshe + ' still in full-time education')

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: NOT (AgeOf IN [19 .. 23])
```

StillEduc := ''

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: AgeOf IN [19 .. 25]
```

**CONTINUOUS := (''CONTINUOUS' CAN INCLUDE A BREAK, IF LESS THAN ' + '18 MONTHS.
CODE '96' IF STILL IN CONTINUOUS F/T ED.')**

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: AgeOf IN [26 .. 120]
```

CONTINUOUS := ('ENTER AGE (OR CODE '96' IF STILL IN ' + 'CONTINUOUS FULL-TIME EDUCATION).')

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: NOT (AgeOf IN [26 .. 120])
```

CONTINUOUS := ''

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
```

TEA

HHG

Include the following as part of 'continuous education':
A 'gap year' between school and college/university, as long as there is/was some clear intention to continue education.
A holiday job during a course, provided they intend to continue with the course.
The working section of a sandwich course.
National Service, if it occurred between school and college/university.

5..97

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: TEA IN [5 .. 95]
     TEA <= AgeOf
```

You've coded that ^LName is ^AgeOf years old, but left full-time education at the age of ^TEA. Please amend the one or the other.

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: (FTEd = No) AND (AgeOf IN [16 .. 18])
     TEA <> 96
```

At the previous question you say that ^LName is not in full-time education. Please amend your answers.

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
```

TEAEx

HHG
This is a 'Key Question': it is very important to get an answer if you possibly can. An ESTIMATE is preferable to Don't know (or refusal).

^SuppTxt

OPEN

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = NONRESPONSE
     ERROR
```

This is a 'Key Question': it is very important to get an answer if you possibly can. An ESTIMATE is preferable to Don't know (or refusal).

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = NONRESPONSE
```

TEAEx

HHG
This is a 'Key Question': it is very important to get an answer if you possibly can. An ESTIMATE is preferable to Don't know (or refusal).

^SuppTxt

OPEN

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = RESPONSE
     AND: AgeOf > 25
     TEA <> 96
```

This person is over 25, so is unlikely to still be in CONTINUOUS full-time education (ie. having been OUT of education for less than 18 months). Please check.

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = RESPONSE
     (TEA >= 14) OR (TEA = 97)
```

This value seems low.
Please check that it is correct.

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = RESPONSE
     AND: TEA < 96
     TEA < 28
```

This value seems high.
Please check that it is correct.

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
```

TypeEd

HHG
Secondary Schools include: Secondary Modern, Grammar, Comprehensive and Technical Schools.

Age ranges for Middle-deemed Primary:
8-12
9-12
9-13

Age ranges for Middle-deemed Secondary:
9-13
10-13
10-14.

(1) Nursery School/Nursery Class/Playgroup/Pre-school
(2) ^State_run Primary (including reception classes)
(3) Special School ^State_run (e.g. for children with disabilities and special educational needs)
(4) ^Mid_Pri
(5) ^Mid_Sec
(6) Secondary^Grammar school ^assisted
(7) Non-advanced further education/ 6th form/tertiary/further education college
(8) Any PRIVATE/Independent school (prep, primary, secondary, City Technology Colleges)
(9) University/polytechnic/any other higher education
(10) Home Schooling

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     AND: TypeEd IN [Sec .. Nonadv]
     AgeOf >= 8
```

This doesn't sound right in relation to ^LName's age:
Please check your entry.

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     AND: TypeEd = Nursery
     IN(AgeOf,[2..12])
```

This doesn't sound right in relation to ^LName's age:
Please check your entry.

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     AND: (TypeEd = Special) AND (AgeOf IN [16 .. 18])
```

SchChk

HHG

INTERVIEWER: PLEASE CHECK: IS CHILD BENEFIT STILL RECEIVED FOR THIS PERSON? (IF YES, THIS CONFIRMS THEY STILL BELONG TO SOMEONE ELSE'S BENEFIT UNIT).

(1) Yes, child benefit still received
(2) No

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NatCen = NI
     NOT(IN(TypeEd,[???,???]))
```

Interviewer, do not use this code at TypeEd.

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
```

DoB

HHG

May I check, what is ^LName's date of birth?

IF DAY NOT KNOWN, ENTER 15th.

DATE

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
     AND: DoB = RESPONSE
     DoB <= QSignIn.StartDat
```

You've entered a future date!

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
     AND: DoB = RESPONSE
     AND: AgeOf IN [1 .. 19]
     AGE(DoB,QSignIn.StartDat) = AgeOf
```

This date doesn't agree with the age and the date of interview. Please check.

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
     AND: DoB = RESPONSE
     AND: (AgeOf = 0) AND (AgeOf = RESPONSE)
     ((QSignIn.StartDat.JULIAN - DoB.JULIAN) <= 365) AND
     INVOLVING(AgeOf,DoB)
```

This date doesn't agree with the age and the date of interview. Please check.

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

Depend

HHG
Status indicator of whether this adult is treated as dependent.

(1) Independent adult
(2) 16-18 years old AND in F/T education
(3) 0-15 years old

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

LiveWith

HHG
Cohabitee?

(1) Yes
(2) No

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

DVMarDF

HHG
De facto marital status

(1) Married
(2) Cohabiting
(3) Single
(4) Widowed
(5) Divorced
(6) Separated
(7) Same sex couple

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

FRS0405A.HHG (continued)

Data on household members

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

LegGuard[[P1] := 2

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].NumParn := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].NumPart := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].NumCohab := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].Spouses := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].Parent1 := 1

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].Parent2 := 1

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].LiveWith := No

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [Spouse .. Cohabit, Sib .. ILSib, OthRel,
     NonRel]
```

P[P1].QRel[P2].R := P[P2].QRel[P1].R

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [Child .. ILChild]
```

P[P1].QRel[P2].R := (ORD(P[P2].QRel[P1].R) + 4)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [Parent .. ILParent]
```

P[P1].QRel[P2].R := (ORD(P[P2].QRel[P1].R) - 4)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [GChild]
```

P[P1].QRel[P2].R := GParent

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [GParent]
```

P[P1].QRel[P2].R := GChild

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
```

P[P1].LiveWith := Yes

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
     AND: P[P1].Sex = P[P2].Sex
```

P[P1].DVMarDF := SamSex

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
     AND: NOT (P[P1].Sex = P[P2].Sex)
```

P[P1].DVMarDF := Cohab

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Marr
```

P[P1].DVMarDF := Married

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].CupChk = Yes
```

P[P1].DVMarDF := Cohab

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].CupChk = SameSex
```

P[P1].DVMarDF := SamSex

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].AgeOf < 16
```

P[P1].DVMarDF := DFSingle

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].MS = Single) AND (P[P1].LiveWith <> Yes)
```

P[P1].DVMarDF := DFSingle

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Widowed
```

P[P1].DVMarDF := DFWidow

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Divorce
```

P[P1].DVMarDF := DFDivor

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Sep
```

P[P1].DVMarDF := DFSepar

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Spouse
     P[P1].Sex <> P[P2].Sex AND INVOLVING(P[P1].Sex,P[P2].Sex)
```

A married partner must be of opposite sex.

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Spouse
     ((P[P2].MS = Marr) OR P[P2].MS=EMPTY) AND INVOLVING(P[P2].QRel[P1].R)
```

You've recorded ^P[P1].Name as the spouse of ^P[P2].Name who is NOT 'Married & living with spouse'. Please amend one or the other.

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
     (P[P2].MS <> Marr) AND INVOLVING(P[P2].QRel[P1].R)
```

You've recorded ^P[P1].Name as 'cohabiting' with ^P[P2].Name, who is 'MARRIED & living with spouse'. Please amend one or the other

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
     P[P1].Sex <> P[P2].Sex AND INVOLVING(P[P2].QRel[P1].R)
```

A cohabiting partner is usually of opposite sex.

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P2].QRel[P1].R = Spouse
     ((P[P1].MS = Marr) OR P[P1].MS=EMPTY) AND INVOLVING(P[P2].QRel[P1].R)
```

You've recorded ^P[P2].Name as the spouse of ^P[P1].Name, who is not coded as being 'Married & living with spouse'. Please amend one or the other

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P2].QRel[P1].R = Cohabit
     (P[P1].MS <> Marr) AND INVOLVING(P[P2].QRel[P1].R)
```

You've recorded ^P[P2].Name as 'cohabiting' with ^P[P1].Name, who is coded as 'MARRIED & living with spouse'. Please amend one or the other

```
WARN IF: HHSize > 0
    AND: P[FHHSize].Sex = RESPONSE
    AND: In loop FOR P1 := 1 TO FHHSize
    AND: In loop FOR P2 := 1 TO FHHSize
    AND: P[P1].QRel[P2].R IN [Parent .. ILParent, GParent]
    (P[P1].AgeOf > 15) AND INVOLVING(P[P2].QRel[P1].R,P[P1].AgeOf)
```

You've coded ^P[P1].Name as a parent (inc. foster/in-law/step) or grandparent, but he/she is less than 16 years old. Please check ^P[P1].Name's age.

```
WARN IF: HHSize > 0
    AND: P[FHHSize].Sex = RESPONSE
    AND: In loop FOR P1 := 1 TO FHHSize
    AND: In loop FOR P2 := 1 TO FHHSize
    AND: P[P1].QRel[P2].R IN [Child .. ILChild, GChild]
    (P[P1].AgeOf < P[P2].AgeOf) AND INVOLVING(P[P2].QRel[P1].R)
```

Children (inc. foster/in-law/step) and grandchildren should normally be younger than their parents/grandparents/ step-parents. Please check the ages you have entered.

```
WARN IF: HHSize > 0
    AND: P[FHHSize].Sex = RESPONSE
    AND: In loop FOR P1 := 1 TO FHHSize
    AND: In loop FOR P2 := 1 TO FHHSize
    AND: P[P1].QRel[P2].R IN [Parent .. ILParent, GParent]
    (P[P1].AgeOf > P[P2].AgeOf) AND INVOLVING(P[P2].QRel[P1].R)
```

Parents (inc. foster/in-law/step) or grandparents, are normally older than their child/grandchild/step-child. Please check the ages and relationships you've entered.

```
WARN IF: HHSize > 0
    AND: P[FHHSize].Sex = RESPONSE
    AND: In loop FOR P1 := 1 TO FHHSize
    AND: In loop FOR P2 := 1 TO FHHSize
    AND: P[P1].QRel[P2].R IN [Child .. ILChild]
    (P[P1].AgeOf < (P[P2].AgeOf - 12)) AND INVOLVING(P[P2].QRel[P1].R)
```

Children (inc. foster/in-law/step) are normally at least 12 years younger than parents/step-parents. Please check the ages you have entered.

```
WARN IF: HHSize > 0
    AND: P[FHHSize].Sex = RESPONSE
    AND: In loop FOR P1 := 1 TO FHHSize
    AND: In loop FOR P2 := 1 TO FHHSize
    AND: P[P1].QRel[P2].R IN [Parent .. ILParent]
    (P[P1].AgeOf > (P[P2].AgeOf + 12)) AND INVOLVING(P[P2].QRel[P1].R)
```

Parents (inc. foster/in-law/step) are normally at least 12 years older than their child/step-child. Please check the ages and relationships you've entered.

```
WARN IF: HHSize > 0
    AND: P[FHHSize].Sex = RESPONSE
    AND: In loop FOR P1 := 1 TO FHHSize
    AND: In loop FOR P2 := 1 TO FHHSize
    AND: P[P1].QRel[P2].R = GChild
    (P[P1].AgeOf < (P[P2].AgeOf - 24)) AND INVOLVING(P[P2].QRel[P1].R)
```

Children are normally at least 24 years younger than their grandparents. Please check the ages you have entered.

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = GParent
     (P[P1].AgeOf > (P[P2].AgeOf + 24)) AND INVOLVING(P[P2].QRel[P1].R)
```

Grandparents are normally at least 24 years older than their grandchildren. Please check the ages and relationships you've entered.

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Spouse]
```

P[P1].Spouses := (P[P1].Spouses + 1)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Spouse .. Cohabit]
```

P[P1].NumPart := (P[P1].NumPart + 1)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
```

P[P1].NumCohab := (P[P1].NumCohab + 1)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].AgeOf IN [0 .. 15]) AND (P[P1].AgeOf = RESPONSE)
```

P[P1].Depend := Child

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].NumPart > 0
```

P[P1].Depend := Adult

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].AgeOf IN [16 .. 18]) AND ((P[P1].TypeEd = Special) AND
     (P[P1].SchChk = No))
```

P[P1].Depend := Adult

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].AgeOf IN [16 .. 18]) AND (P[P1].TypeEd IN [Special ..
     Private])
```

P[P1].Depend := DepAd

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: NOT ((P[P1].AgeOf IN [16 .. 18]) AND (P[P1].TypeEd IN [Special ..
     Private])
```

P[P1].Depend := Adult

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     (IN(P[P1].NumPart,[0..1])) AND
     INVOLVING(P[P2].QRel[P1].R,P[P1].QRel[P2].R)
```

^P[P1].Name has more than one spouse/cohabitee. Establish who is principal partner, & re-code the other as '17' or '18'.

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
     AND: P[P1].Parent1 = EMPTY
```

P[P1].Parent1 := P2

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
     AND: P[P1].Parent2 = EMPTY AND (P[P1].Parent1 <> P2)
     P[P[P1].Parent1].Sex <> P[P2].Sex AND INVOLVING(P[P2].QRel[P1].R)
```

The parents of ^P[P1].Name are of the same sex. Please check.

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
     AND: P[P1].Parent2 = EMPTY AND (P[P1].Parent1 <> P2)
```

P[P1].Parent2 := P2

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
```

P[P1].NumParn := (P[P1].NumParn + 1)

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = Adult) AND INVOLVING (P[P2].QRel[P1].R,
     P[P1].QRel[P2].R)
     P[P1].NumParn <= 2
```

This suggests that ^P[P1].Name has more than two parents. Please check the relationship codes for ^P[P1].Name and select which one to alter.

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = DepAd) AND INVOLVING (P[P2].QRel[P1].R,
     P[P1].QRel[P2].R)
     P[P1].NumParn <= 2
```

You've given ^P[P1].Name more than two parents (inc step/foster). To calculate Benefit Units properly you must reduce this to a maximum of two. Select which one to alter, and re-code as '17'. (Check who receives Child Benefit for ^P[P1].Name).

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = Child) AND (P[P1].QRel[P2].R IN [Child .. FChild])
```

LegGuard[[P1] := 1

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = DepAd) AND (P[P1].QRel[P2].R IN [Spouse ..
     FChild])
```

LegGuard[[P1] := 1

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: (P[P1].Depend = DepAd) AND (P[PHHSize].QRel[PHHSize - 1].R =
     RESPONSE)
     (LegGuard[[P1] = 1) AND INVOLVING(P[P1].QRel[1].R)
```

Who in the household is responsible for ^P[P1].Name - is there a legal guardian, or does anyone get Child Benefit for ^P[P1].Name? If so, recode ^P[P1].Name as that person's legal dependent (Code 3) or that person as ^P[P1].Name's parent (code 7). If not, suppress warning and continue.

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: P[P1].Sex = Male
```

himher := 'him'

```
COMPUTE IF: HHSize > 0
      AND: P[FHHSize].Sex = RESPONSE
      AND: In loop FOR P1 := 1 TO FHHSize
      AND: P[P1].Sex = Female
```

himher := 'her'

```
CHECK IF: HHSize > 0
      AND: P[FHHSize].Sex = RESPONSE
      AND: In loop FOR P1 := 1 TO FHHSize
      AND: (P[P1].Depend = Child) AND (P[PHHSize].QRel[PHHSize - 1].R =
      RESPONSE)
      (LegGuard[[P1] = 1) AND INVOLVING(P[P1].AgeOf)
```

^P[P1].Name is under 16, so you MUST recode ^himher as the child (relationship code 3) of an adult: in order of priority, the person receiving Child Benefit for ^himher, or the legal guardian, or whoever is responsible for ^himher.

(NB. If no-one in the household is over age 15, the household is ineligible - outcome code ^Out772).

```
CHECK IF: HHSize > 0
      AND: P[FHHSize].Sex = RESPONSE
      AND: In loop FOR P1 := 1 TO FHHSize
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
      AND: P[FHHSize].Sex = RESPONSE
      AND: In loop FOR P1 := 1 TO FHHSize
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
      RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

CHECK IF: *HHSize > 0*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)*
AND: *SpOut[Loop1] = Amend*
HHG.P[Loop1].Sex <> RESPONSE

Press <Enter> to return to the household grid.

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

AllNameNo := ''

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

AdNameNo := ''

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

AllAd := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

AllCh := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

ChUnder1 := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

ChRegis := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

Over75 := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

Over60 := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

Under19 := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *HHG.P[Loop1].Sex = RESPONSE*

```
AllNameNo := (AllNameNo + '
' + STR(Loop1,2,0) + '. ' + QNames.M[Loop1].Name)
```

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *HHG.P[Loop1].Sex = RESPONSE*
AND: *HHG.P[Loop1].AgeOf IN [16 .. 120]*

```
AdNameNo := (AdNameNo + '
' + STR(Loop1,2,0) + '. ' + QNames.M[Loop1].Name)
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [16 .. 120]
```

OneHRP := Loop1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [16 .. 120]
```

AllAd := (AllAd + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: NOT (HHG.P[Loop1].AgeOf IN [16 .. 120])
```

AllCh := (AllCh + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [0 .. 1]
```

ChUnder1 := (ChUnder1 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf >= 75
```

Over75 := (Over75 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf >= 60
```

Over60 := (Over60 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [0 .. 18]
```

Under19 := (Under19 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMAge[Loop1] := HHG.P[Loop1].AgeOf

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMTEA[Loop1] := HHG.P[Loop1].TEA

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMParent1[Loop1] := HHG.P[Loop1].Parent1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMParent2[Loop1] := HHG.P[Loop1].Parent2

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMNumParn[Loop1] := HHG.P[Loop1].NumParn

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].Sex := HHG.P[Loop1].Sex

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].MS := HHG.P[Loop1].MS

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].W1 := HHG.P[Loop1].W1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].W2 := HHG.P[Loop1].W2

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].FtEd := HHG.P[Loop1].FTEd

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].TypeEd := HHG.P[Loop1].TypeEd

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].Depend IN [DepAd .. Child]
```

NCDVCP := (NCDVCP + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: (HHG.P[Loop1].Depend IN [DepAd .. Child]) AND
     ((HHG.P[Loop1].NumPart > 0) OR (LegGuard[Loop1] = 2))
```

PRec[Loop1].Depend := Adult

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: NOT ((HHG.P[Loop1].Depend IN [DepAd .. Child]) AND
     ((HHG.P[Loop1].NumPart > 0) OR (LegGuard[Loop1] = 2))
```

PRec[Loop1].Depend := HHG.P[Loop1].Depend

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: (HHG.P[Loop1].CupChk IN [Yes, SameSex]) AND
     HHG.P[HHSize].QRel[HHSize].R <> EMPTY
     (HHG.P[Loop1].NumCohab > 0) AND
     INVOLVING(HHG.P[HHSize].QRel[HHSize].R,HHG.P[Loop1].CupChk)
```

(^HHG.P[Loop1].Name) Can I just check, you said earlier that you were living with someone in this household as a couple, is this correct?

IF YES: go back and correct the code at Relationship.
IF NO: go back and correct CupChk to No (2).

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: (HHG.P[Loop1].CupChk = No) AND HHG.P[HHSize].QRel[HHSize].R <>
     EMPTY
     (HHG.P[Loop1].NumCohab = 0) AND
     INVOLVING(HHG.P[HHSize].QRel[HHSize].R,HHG.P[Loop1].CupChk)
```

(^HHG.P[Loop1].Name) Can I just check, are you living with anyone in the household as a couple?

IF YES: go back and correct CupChk to Yes (1).
IF NO: go back and correct the code at Relationship.

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: AllAd = 1
```

you := 'you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: NOT (AllAd = 1)
```

you := 'ANY of you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (AllAd + AllCh) = 1
```

any_of_you := 'you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: NOT ((AllAd + AllCh) = 1)
```

any_of_you := 'ANY of you'

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

RentName := ''

FRS0405A.QHholder

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[1] := ' 1. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[2] := ' 2. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[3] := ' 3. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[4] := ' 4. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[5] := ' 5. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[6] := ' 6. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[7] := ' 7. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[8] := ' 8. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[9] := ' 9. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[10] := '10. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[11] := '11. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[12] := '12. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[13] := '13. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[14] := '14. '

***Ask if:** HHG.P[HHSize].AgeOf = RESPONSE*

HHldr

In whose name is the accommodation owned or rented?
Anyone else?

CODE ALL THAT APPLY.

SET [15] OF

(1)	^DMName[1]
(2)	^DMName[2]
(3)	^DMName[3]
(4)	^DMName[4]
(5)	^DMName[5]
(6)	^DMName[6]
(7)	^DMName[7]
(8)	^DMName[8]
(9)	^DMName[9]
(10)	^DMName[10]
(11)	^DMName[11]
(12)	^DMName[12]
(13)	^DMName[13]
(14)	^DMName[14]
(97)	Not a household member

***Compute if:** HHG.P[HHSize].AgeOf = RESPONSE*

HhldList := ''

***Compute if:** HHG.P[HHSize].AgeOf = RESPONSE*

HhldNum := 0

***Compute if:** HHG.P[HHSize].AgeOf = RESPONSE*
***And:** NotHH IN HHldr*

HhldCard := (HHldr.CARDINAL - 1)

***Compute if:** HHG.P[HHSize].AgeOf = RESPONSE*
***And:** NOT (NotHH IN HHldr)*

HhldCard := HHldr.CARDINAL

***Compute if:** HHG.P[HHSize].AgeOf = RESPONSE*
***And:** In loop FOR Loop := 1 TO 14*

HhldName[Loop] := ''

FRS0405A.QHholder.PadString()

Procedure Call

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
```

OutString := InString

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: ResLngth > LEN (OutString)
```

Fin := (ResLngth - LENGTH(OutString))

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: ResLngth > LEN (OutString)
     AND: In loop FOR PLoop := 1 TO Fin
```

OutString := (OutString + ' ')

FRS0405A.QHholder (continued)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
```

```
HhldName[Loop] := ('
' + LPad[Loop] + PadName + '  (Age: ' + STR(DMAge[[Loop],3,0) +
')')
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
```

```
HhldNum := (HhldNum + 1)
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: HhldNum = 1
```

```
HhldList := DMName[[Loop]
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: HhldNum < HhldCard
```

```
HhldList := (HhldList + ', ' + DMName[[Loop])
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: HhldNum = HhldCard
```

```
HhldList := (HhldList + ' and ' + DMName[[Loop])
```

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     RESERVECHECK
```

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
RESERVECHECK

RESERVECHECK

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

WhoResp

Although you have mentioned that the rent or mortgage for this accommodation is paid for by someone outside the household, there needs to be someone within the household who is responsible for the property. Who then in this household is responsible for this accommodation?
Anyone else?

CODE ALL THAT APPLY.

SET [14] OF
(1) ^DMName[1]
(2) ^DMName[2]
(3) ^DMName[3]
(4) ^DMName[4]
(5) ^DMName[5]
(6) ^DMName[6]
(7) ^DMName[7]
(8) ^DMName[8]
(9) ^DMName[9]
(10) ^DMName[10]
(11) ^DMName[11]
(12) ^DMName[12]
(13) ^DMName[13]
(14) ^DMName[14]

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

HhldList := ''

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

HhldNum := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

HhldCard := WhoResp.CARDINAL

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
AND: *In loop FOR Loop := 1 TO 14*

HhldName[Loop] := ''

FRS0405A.QHholder.PadString()

Procedure Call

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
```

`OutString := InString`

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: ResLngth > LEN (OutString)
```

`Fin := (ResLngth - LENGTH(OutString))`

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: ResLngth > LEN (OutString)
     AND: In loop FOR PLoop := 1 TO Fin
```

`OutString := (OutString + ' ')`

FRS0405A.QHholder (continued)

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
    AND: *In loop FOR Loop := 1 TO 14*
    AND: *Loop IN WhoResp*

**HhldName[Loop] := ('
' + LPad[Loop] + PadName + ' (Age: ' + STR(DMAge[[Loop],3,0) + ')')**

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
    AND: *In loop FOR Loop := 1 TO 14*
    AND: *Loop IN WhoResp*

HhldNum := (HhldNum + 1)

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
    AND: *In loop FOR Loop := 1 TO 14*
    AND: *Loop IN WhoResp*
    AND: *HhldNum = 1*

HhldList := DMName[[Loop]

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
    AND: *In loop FOR Loop := 1 TO 14*
    AND: *Loop IN WhoResp*
    AND: *HhldNum < HhldCard*

HhldList := (HhldList + ', ' + DMName[[Loop])

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
    AND: *In loop FOR Loop := 1 TO 14*
    AND: *Loop IN WhoResp*
    AND: *HhldNum = HhldCard*

HhldList := (HhldList + ' and ' + DMName[[Loop])

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
    RESERVECHECK

RESERVECHECK

RECORD IF: *HHG.P[HHSize].AgeOf = RESPONSE*

HRPPrtnr

Person number of HRP's spouse/partner.

1..15

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(HHldr <> EMPTY AND (AllAd = 1)) AND (OneHRP = RESPONSE)*

HRPPrtnr := 15

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    RESERVECHECK

RESERVECHECK

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
    AND: *WhoResp.CARDINAL > 1*

own := 'are responsible for'

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
    AND: *NOT (WhoResp.CARDINAL > 1)*

own := 'own or rent'

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*

HiHNum

IF RESPONDENT ASKS FOR PERIOD TO AVERAGE OVER - ONE YEAR.

PROMPT AS NECESSARY FOR JOINT HOUSEHOLDERS:
IS ONE OF THEM THE SOLE PERSON WITH PAID WORK OR OCCUPATIONAL PENSION?

1..15

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
AND: *WhoResp = EMPTY*
AND: *HiHNum IN [1 .. 14]*
(IN(HiHNum,HHldr)) AND INVOLVING(HiHNum)

This person is not recorded as a householder (at HHldr).

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
AND: *NOT (WhoResp = EMPTY)*
AND: *HiHNum IN [1 .. 14]*
(IN(HiHNum,WhoResp)) AND INVOLVING(HiHNum)

This person is not recorded as responsible for the household (at WhoResp).

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
AND: *HiHNum = 15*

JntEldA

ENTER PERSON NUMBER OF THE ELDEST JOINT HOUSEHOLDER FROM THOSE WITH THE SAME HIGHEST INCOME.

ASK OR RECORD.
^HhldName[1]^HhldName[2]^HhldName[3]^HhldName[4]^HhldName[5]
^HhldName[6]^HhldName[7]^HhldName[8]^HhldName[9]^HhldName[10]
^HhldName[11]^HhldName[12]^HhldName[13]^HhldName[14]

0..14

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
AND: *HiHNum = 15*
AND: *JntEldA = RESPONSE*
JntEldA <> 0

Zero (0) is not a valid code.

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
AND: *HiHNum = 15*
AND: *JntEldA IN [1 .. 14]*
AND: *WhoResp = EMPTY*
(IN(JntEldA,HHldr)) AND INVOLVING(HiHNum)

This person is not recorded as a householder (at HHldr).

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
     AND: JntEldA IN [1 .. 14]
     AND: NOT (WhoResp = EMPTY)
     (IN(JntEldA,WhoResp)) AND INVOLVING(HiHNum)
```

This person is not recorded as responsible for the household (at WhoResp).

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
     AND: JntEldA IN [1 .. 14]
```

DVHRPNum := JntEldA

```
ASK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
```

JntEldB

ENTER PERSON NUMBER OF THE ELDEST JOINT HOUSEHOLDER.

ASK OR RECORD.
^HhldName[1]^HhldName[2]^HhldName[3]^HhldName[4]^HhldName[5]
^HhldName[6]^HhldName[7]^HhldName[8]^HhldName[9]^HhldName[10]
^HhldName[11]^HhldName[12]^HhldName[13]^HhldName[14]

0..14

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB = RESPONSE
     JntEldB <> 0
```

Zero (0) is not a valid code.

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB IN [1 .. 14]
     AND: WhoResp = EMPTY
     (IN(JntEldB,HHldr)) AND INVOLVING(HiHNum)
```

This person is not recorded as a householder (at HHldr).

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB IN [1 .. 14]
     AND: NOT (WhoResp = EMPTY)
     (IN(JntEldB,WhoResp)) AND INVOLVING(HiHNum)
```

This person is not recorded as responsible for the household (at WhoResp).

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB IN [1 .. 14]
```

DVHRPNum := JntEldB

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum IN [1 .. 14]
```

DVHRPNum := HiHNum

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND NOT (NotHH IN HHldr)
```

DVHRPNum := ORD(HHldr[1])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 2) AND (NotHH IN HHldr)
     AND: HHldr[1] = NotHH
```

DVHRPNum := ORD(HHldr[2])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 2) AND (NotHH IN HHldr)
     AND: NOT (HHldr[1] = NotHH)
```

DVHRPNum := ORD(HHldr[1])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: WhoResp.CARDINAL = 1
```

DVHRPNum := ORD(WhoResp[1])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: NOT (WhoResp.CARDINAL = 1)
```

DVHRPNum := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: DVHRPNum IN [1 .. 14]
```

LName := DMName[[DVHRPNum]

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: NOT (DVHRPNum IN [1 .. 14])
```

LName := 'Non-HH Member'

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*

HRP

THE HOUSEHOLD REFERENCE PERSON IS:

(^DVHRPNum) ^LName

PRESS 1 AND <ENTER> TO CONTINUE.

(1) Continue

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.HHldr
     PRec[Loop1].Sex = RESPONSE
```

Code ^Loop1 is not valid for this question.

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.HHldr
     PRec[Loop1].Depend = Adult
```

Person ^Loop1 is a child or a dependent adult. Please amend.

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.HHldr
```

HHG.P[Loop1].Hholder := Yes

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: NOT (Loop1 IN QHholder.HHldr)
```

HHG.P[Loop1].Hholder := No

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.WhoResp
     PRec[Loop1].Sex = RESPONSE
```

This figure exceeds the number of household members. Please check and amend your answer.

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.WhoResp
     PRec[Loop1].Depend = Adult
```

The person responsible for the property must be an adult household member. Please check and amend your answer.

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     PRec[QHholder.DVHRPNum].Depend = Adult
```

Code ^QHholder.DVHRPNum is not valid for this question.

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     NOT((PRec[QHholder.DVHRPNum].Sex = Female) AND
     (PRec[QHholder.DVHRPNum].MS = Marr))
```

For a married couple the man is always Head of household.
Please amend your coding. (But if he is away for more than 6 months, suppress check and move on.)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
```

HRPNames := DMName[QHholder.DVHRPNum]

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
```

PRel.PR[Loop1].R := HHG.P[Loop1].QRel[QHholder.DVHRPNum].R

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].QRel[QHholder.DVHRPNum].R IN [Spouse, Cohabit]
```

HRPNames := (HRPNames + ' and ' + DMName[Loop1])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].QRel[QHholder.DVHRPNum].R IN [Spouse, Cohabit]
```

QHholder.HRPPrtnr := Loop1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: Loop1 = QHholder.DVHRPNum
```

QHholder.QPerId[Loop1].HOHID := HOH

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: Loop1 = QHholder.DVHRPNum
```

QHholder.QPerId[Loop1].HRPID := HRP

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: Loop1 = QHholder.DVHRPNum
```

QHholder.QPerId[Loop1].CombID := HOHHRP

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: NOT (Loop1 = QHholder.DVHRPNum)
```

QHholder.QPerId[Loop1].HOHID := NotHOH

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: NOT (Loop1 = QHholder.DVHRPNum)
```

QHholder.QPerId[Loop1].HRPID := NotHRP

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: NOT (Loop1 = QHholder.DVHRPNum)
```

QHholder.QPerId[Loop1].CombID := HOHonly

FRS0405A.QEthnic

Ethnic data on adults in household

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

P[Personnr].PersId := Personnr

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

P[Personnr].BenUnit := DMBU[[Personnr]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

P[Personnr].EName := DMName[[Personnr]

FRS0405A.QEthnic.P[]

```
RECORD IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

BenUnit

QEthnic

Benefit Unit number.

1..7

```
RECORD IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

PersId

QEthnic

Person identifier.

0..14

```
DISPLAY IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

EName

QEthnic

STRING[15]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

LName := EName

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: PRec[PersId].Sex = Male
```

he_she := 'he'

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (PRec[PersId].Sex = Male)
```

he_she := 'she'

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
```

NatID

QEthnic

SHOW CARD B
*(^LName) What do you consider your national identity to be? Please choose your answer from this card, choose as many or as few as apply.
PROBE: Any other?

SET [6] OF
(1) English
(2) Scottish
(3) Welsh
(4) Irish
(5) British
(6) Other answer

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
     AND: Other IN NatID
```

NatOth

QEthnic

(^LName)
How would you describe your national identity?

INTERVIEWER INSTRUCTION - If someone describes themselves as being half English and half Irish or any combination of Welsh, Scottish, Irish or English, code them as 'Mixed British' AND THEN RECORD THE MIX THEY SPECIFY.

(1) Mixed British
(2) ENTER DESCRIPTION OF ETHNIC GROUP

Ask if: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
And: *In loop FOR Personnr := 1 TO HHSize*
And: *PRec[Personnr].Depend = Adult*
And: *NatCen <> NI*
And: *Other IN NatID*

XNatOth

QEthnic

(^LName)
ENTER DESCRIPTION OF ETHNIC GROUP.

STRING[100]

Ask if: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
And: *In loop FOR Personnr := 1 TO HHSize*
And: *PRec[Personnr].Depend = Adult*
And: *NatCen <> NI*

EthGrp

QEthnic

SHOW CARD C
To which of these ethnic groups does ^LName consider ^he_she belongs?
THIS IS A QUESTION OF RESPONDENT'S (OR PROXY'S) OPINION.

(1) White - British
(2) Any other white background (please describe)
(3) Mixed - White and Black Caribbean
(4) Mixed - White and Black African
(5) Mixed - White and Asian
(6) Any other mixed background (please describe)
(7) Asian or Asian British - Indian
(8) Asian or Asian British - Pakistani
(9) Asian or Asian British - Bangladeshi
(10) Any other Asian/Asian British background (please describe)
(11) Black or Black British - Caribbean
(12) Black or Black British - African
(13) Any other Black/Black British background (please describe)
(14) Chinese
(15) Any other (please describe)

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
     AND: EthGrp IN [WhtOth, MixedOth, AsianOth, BlackOth, Other]
```

EthOth

QEthnic

(^LName)
Please can you describe your ethnic group?

ENTER DESCRIPTION OF ETHNIC GROUP.

STRING[100]

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (NatCen <> NI)
```

NINatID

QEthnic

SHOW CARD B
*(^LName) What do you consider your national identity to be? Please choose your answer from this card, choose as many or as few as apply.
PROBE: Any other?

SET [8] OF
(1) British
(2) Irish
(3) Ulster
(4) Northern Irish
(5) English
(6) Scottish
(7) Welsh
(8) Other answer

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (NatCen <> NI)
     AND: Other IN NINatID
```

NINatOth

QEthnic

(^LName)
How would you describe your national identity?

INTERVIEWER INSTRUCTION - If someone describes themselves as being half English and half Irish or any combination of Welsh, Scottish, Irish or English, code them as 'Mixed British' AND THEN RECORD THE MIX THEY SPECIFY.

(1) Mixed British
(2) ENTER DESCRIPTION OF ETHNIC GROUP

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (NatCen <> NI)
     AND: Other IN NINatID
```

NIXNatOth

QEthnic

(^LName)
ENTER DESCRIPTION OF ETHNIC GROUP.

STRING[100]

ASK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*
AND: *NOT (NatCen <> NI)*

NIEthGrp

QEthnic

SHOW CARD C
To which of these ethnic groups does ^LName consider ^he_she belongs?
THIS IS A QUESTION OF RESPONDENT'S (OR PROXY'S) OPINION.

(1) White
(2) Irish Traveller
(3) Any other white background (please describe)
(4) Mixed - White and Black Caribbean
(5) Mixed - White and Black African
(6) Mixed - White and Asian
(7) Any other mixed background (please describe)
(8) Asian - Indian
(9) Asian - Pakistani
(10) Asian - Bangladeshi
(11) Any other Asian background (please describe)
(12) Black - Caribbean
(13) Black - African
(14) Any other Black background (please describe)
(15) Chinese
(16) Any other (please describe)

ASK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*
AND: *NOT (NatCen <> NI)*
AND: *NIEthGrp IN [WhtOth, AsianOth, BlackOth, Other]*

NIEthOth

QEthnic

(^LName)
Please can you describe your ethnic group?

ENTER DESCRIPTION OF ETHNIC GROUP.

STRING[100]

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*
RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*
RESERVECHECK

RESERVECHECK

```
WARN IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     RESERVECHECK
```

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend IN [DepAd .. Child]*
AND: *((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])*

**DepParnt := (STR(DMParent1[Loop1],1,0) + ' : ' + DMName[DMParent1[Loop1]] + '
' + STR(DMParent2[Loop1],1,0) + ' : ' + DMName[DMParent2[Loop1]])**

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend IN [DepAd .. Child]*
AND: *((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])*
AND: *PRec[Loop1].Sex = Male*

HeShe := 'HE'

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend IN [DepAd .. Child]*
AND: *((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])*
AND: *PRec[Loop1].Sex = Male*

HisHer := 'HIS'

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])
     AND: NOT (PRec[Loop1].Sex = Male)
```

HeShe := 'SHE'

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])
     AND: NOT (PRec[Loop1].Sex = Male)
```

HisHer := 'HER'

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])
```

LegDep

INTERVIEWER: ^DMName[Loop1] IS CLASSIFIED AS A DEPENDANT ADULT OR A CHILD, ie. ^HeShe WILL NOT FORM A Benefit Unit OF ^HisHer OWN.
TO PROPERLY ASSESS TO WHICH Benefit Unit ^HeShe BELONGS, PLEASE CODE WHICH OF THE PARENTS RECEIVE Child Benefit FOR ^DMName[Loop1].

^DepParnt

1..97

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])
     AND: LegDep[Loop1] = RESPONSE
     (LegDep[Loop1] = DMParent1[Loop1]) OR (LegDep[Loop1] =
     DMParent2[Loop1])
```

Code ^LegDep[Loop1] is not valid for this question.

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
```

ABen[Loop1] := 1

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

RESERVECHECK

RESERVECHECK

RECORD IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

NewBU

Total number of BUs

0..7

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

ABen[QHholder.DVHRPNum] := 1

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

Last := 1

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [Adult])*
AND: *In loop FOR Loop2 := 1 TO HHSize*
AND: *(ABen[Loop2] <> EMPTY AND (HHG.P[Loop1].QRel[Loop2].R IN [Spouse, Cohabit])) AND PRec[Loop1].Sex <> PRec[Loop2].Sex*

ABen[Loop1] := ABen[Loop2]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)*
AND: *Last < 7*

Last := (Last + 1)

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)*
AND: *Last < 7*

ABen[Loop1] := Last

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)*
AND: *NOT (Last < 7)*

ABen[Loop1] := 0

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])
     AND: LegDep[Loop1] = RESPONSE
```

ABen[Loop1] := ABen[LegDep[Loop1]]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])
     AND: DMParent1[Loop1] <> 0
```

ABen[Loop1] := ABen[DMParent1[Loop1]]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])
     AND: DMParent2[Loop1] <> 0
```

ABen[Loop1] := ABen[DMParent2[Loop1]]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])
     AND: NOT (DMParent2[Loop1] <> 0)
     AND: Last < 7
```

Last := (Last + 1)

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])
     AND: NOT (DMParent2[Loop1] <> 0)
     AND: Last < 7
```

ABen[Loop1] := Last

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])
     AND: NOT (DMParent2[Loop1] <> 0)
     AND: NOT (Last < 7)
```

ABen[Loop1] := 0

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
```

NewBU := Last

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*

NameInBU[Loop1] := ''

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend IN [Adult]*

NameInBU[ABen[Loop1]] := (NameInBU[ABen[Loop1]] + UPCASE(DMName[Loop1]) + ' ')

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend IN [DepAd .. Child]*

NameInBU[ABen[Loop1]] := (NameInBU[ABen[Loop1]] + DMName[Loop1] + ' ')

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO NewBU*

NameInBU[Loop1] := ('
' + STR(Loop1,1,0) + ': ' + NameInBU[Loop1])

ASK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

ShowBen

If you think that the computer has made a mistake in allocating the individuals to Benefit Units, go back to the household grid and check the relationship codes of each person.

1..1

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
```

hhchu11 := No

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Sex = RESPONSE
```

DMBU[Loop1] := ABen[Loop1]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Sex = RESPONSE
```

HHG.P[Loop1].BenUnit := ABen[Loop1]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: (DMBU[Loop1] = 1) AND (DMAge[Loop1] IN [0 .. 10])
```

hhchu11 := Yes

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: Loop1 IN QHholder.HHldr
```

RentName := (RentName + ' ' + DMName[Loop1])

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: Loop1 IN QHholder.HHldr
     AND: DMBU[Loop1] <> 1
```

NotHRPBU := 1

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend = Adult
     AND: BUAdName[DMBU[Loop1]] =
```

BUAdName[DMBU[Loop1]] := DMName[Loop1]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend = Adult
     AND: NOT (BUAdName[DMBU[Loop1]] =)
```

BUAdName[DMBU[Loop1]] := (BUAdName[DMBU[Loop1]] + ' and ' + DMName[Loop1])

FRS0405A.QAccomdat

Questions about accommodation

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Ten1Ex

QAccomDat

^KeyTxt

^SuppTxt

OPEN

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Ten2Rs

QAccomDat

Can I just check do you live rent free because you receive 100% housing benefit?
INTERVIEWER: IF YES Recode Tenure to Code 4 (Rent it).

PLEASE CHECK THEIR RENT/MORTGAGE IS NOT PAID BY BENEFITS. ONLY ACCOMMODATION PROVIDED BY SOMEONE ELSE (EMPLOYER, RELATIVE, ETC) IS RENT-FREE.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Ten2Ex

QAccomDat

^SuppTxt

OPEN

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Tenure

QAccomDat

SHOW CARD D
In which of these ways do you occupy this accommodation?

(1) Own it outright
(2) Buying it with the help of a mortgage or loan
(3) Pay part rent and part mortgage (^SharOwn)
(4) Rent it
(5) Live here rent-free (including in a relative's/friend's property; excluding squatting)
(6) Squatting

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Tenure = NONRESPONSE*
ERROR

^KeyTxt

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Tenure = NONRESPONSE*

Ten1Ex

QAccomDat

^KeyTxt

^SuppTxt

OPEN

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
Tenure <> RentFree

Tenure <> RentFree

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Ten2Rs = Suppressed*

Ten2Ex

QAccomDat

^SuppTxt

OPEN

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Tenure = Part*

SOBuy

QAccomDat

INTERVIEWER, ASK OR CODE:

^SOWNERS: Are you still buying your share in this (house/flat), or have you now paid off that mortgage or loan?

(1) Still buying
(2) Mortgage is paid off

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

SubLet

QAccomDat

Do you have a formal arrangement to let, or sub-let, any part of this accommodation to someone who is NOT a member of your household?

(1) Yes
(2) No

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *SubLet = Yes*

```
How := ('Thinking just of the accommodation occupied ' + 'by
your household,
how')
```

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *SubLet = Yes*

SubLetY

QAccomDat

Who is that? CODE FIRST THAT APPLIES.

INTERVIEWER: CLOSE RELATIVES = Householder's PARTNER,
PARENT (incl.STEP-), SON or DAUGHTER (incl. STEP-),
BROTHER or SISTER, or SPOUSE of any of these.

(1) Close relative
(2) Other relative
(3) Non-relative

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *NOT (SubLet = Yes)*

```
How := 'How'
```

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Rooms

QAccomDat

In general, include any room which is habitable or usable by the household all year round.

If a room is open-plan count it as 2 rooms if it is divided by a fixed sliding or folding partition.

0..20

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

RoomShar

QAccomDat

Are any of these rooms shared with anyone who is not a member of your household?

IF 'NO' ENTER '0'.
IF 'YES', ASK: How many? AND ENTER NUMBER.

0..10

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RoomShar <> 2

The answer you have entered means two rooms are shared. If you intended to answer 'No' to this question, please change the code to '0' (zero). Otherwise, suppress this warning.

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Bedroom

QAccomDat

^How many bedrooms do you have in this accommodation?

INCLUDE ANY ROOM USED FOR SLEEPING.

1..10

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*

ChkTxt := ('cannot be greater than total number of rooms: ' + 'please check your answers and amend as necessary.')

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RoomShar <= Rooms

Number of shared rooms ^ChkTxt

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
Bedroom <= Rooms

Number of bedrooms ^ChkTxt

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

MainAcc

QAccomDat

Note that this is accommodation occupied by the household. If the household occupies a flat in a converted house, code as a flat.

(1) a house or bungalow
(2) a flat or maisonette
(3) a room or rooms
(4) or something else?

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Shelter

QAccomDat

Is this sheltered accommodation?

INTERVIEWER: HOUSING WITH A WARDEN AND/OR ALARMS.

(1) Yes
(2) No

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

Detach := 'detached'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

SemiDetach := 'semi-detached'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

Terrace := 'or terraced/end of terrace?'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

PurposeBuilt := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

ConvertedHouse := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

MobileHome := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

OtherKind := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

Accommodation := 'the house or bungalow'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

Detach := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

SemiDetach := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

Terrace := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

PurposeBuilt := 'a purpose-built block'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

ConvertedHouse := 'or a converted house/some other kind of building?'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

MobileHome := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

OtherKind := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

Accommodation := 'the flat/maisonette'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

Detach := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

SemiDetach := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

Terrace := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

PurposeBuilt := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

ConvertedHouse := '<NOT USED>'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

MobileHome := 'a caravan, mobile home or houseboat'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

OtherKind := 'or some other kind of accommodation?'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

Accommodation := 'the accommodation'

```
ASK IF: HHG.P[HHSize].BenUnit = RESPONSE
    AND: MainAcc IN [HseBun .. FltMas, Oth]
```

TypeAcc

QAccomDat

Houses which are joined only by a garage (link-detached) should be coded detached.

(1) ^Detach
(2) ^SemiDetach
(3) ^Terrace
(4) ^PurposeBuilt
(5) ^ConvertedHouse
(6) ^MobileHome
(7) ^OtherKind

```
CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
    AND: MainAcc IN [HseBun .. FltMas, Oth]
    AND: TypeAcc = RESPONSE
    (((IN(TypeAcc,[???])) AND (MainAcc = HseBun)) OR
    ((IN(TypeAcc,[???,???])) AND (MainAcc = FltMas))) OR ((MainAcc = Oth)
    AND (IN(TypeAcc,[???])))
```

This code is not valid for this accommodation.

```
CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
    AND: MainAcc IN [HseBun .. FltMas, Oth]
    AND: TypeAcc IN [Detached .. Purpose_built]
    (SubLet <> Yes) AND INVOLVING(MainAcc)
```

As part of this accommodation is sub-let, this household cannot be coded as occupying a whole house, flat etc.
Use another code at MainAcc instead.

```
ASK IF: HHG.P[HHSize].BenUnit = RESPONSE
    AND: (TypeAcc IN [Purpose_built, Converted_house]) OR (MainAcc = ARoom)
```

Floor

QAccomDat

What is the floor level of this household's accommodation?

(1) Basement/semi-basement
(2) Ground floor/street level
(3) 1st floor (floor above street level)
(4) 2nd floor
(5) 3rd floor
(6) 4th Floor
(7) 5th to 9th floor
(8) 10th floor or higher
(9) Don't Know

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Entry

QAccomDat

Are there any physical barriers to entry to the house/flat/accommodation?

CODE ALL THAT APPLY

SET [4] OF
(1) Locked common entrance
(2) Locked gates
(3) Security staff or other gatekeeper
(4) Entry phone access
(5) None

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *None IN Entry*
Entry.CARDINAL = 1

'None' is an exclusive code.

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

YearLive

QAccomDat

For how many years have you, (that is ^PHRPName), lived at this address?

PROBE TO CLASSIFY.

(1) Less than 12 months
(2) 12 months but less than 2 years
(3) 2 years but less than 3 years
(4) 3 years but less than 5 years
(5) 5 years but less than 10 years
(6) 10 years but less than 20 years
(7) 20 years or longer

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *YearLive = Less12m*

MonLive

QAccomDat

For how many months have you, (that is ^PHRPName), lived at this address?

ENTER NUMBER OF MONTHS, TO NEAREST WHOLE MONTH.

0..11

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *NewBU = 1*

HHStat := Conv

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
  AND: *NOT (NewBU = 1)*

HHStat

QAccomDat

INTERVIEWER : CLASSIFY THIS HOUSEHOLD AS ONE OF THE FOLLOWING:
NOTE: Conventional Households include:
- tenure is owner occupier and a 2nd or 3rd benefit unit is paying rent
- tenure is rent free but 2nd or 3rd BU receives Housing Benefit.
- 2rd or 3rd BU members paying rent to the householder(s) in BU1 are also named as householders (this is similar to part owned/part rented tenure).

(1) Conventional household: ie. single person or couple - with other family and/or boarder(s) and/or lodger(s)
(2) 'Shared' household arrangements: identity of HRP is unclear or arbitrary - eg. students, nurses, unrelated adults etc, sharing ON EQUAL BASIS

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

AnyVeh

QAccomDat

Do you at present own or have continuous use of any motor vehicles?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

VehNumb

QAccomDat

Number of vehicles.

0..8

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

AdultH

QAccomDat

Actual number of adults in household.

0..14

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

DepChldH

QAccomDat

Actual number of children in household.

0..14

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

DatYrAgo

QAccomDat

Date one year ago

DATE

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

BenUnits

QAccomDat

Actual number of Benefit Units in household.

0..7

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

Premium

QAccomDat

Any insurance policies?

(1) Yes
(2) No

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

Dentist

QAccomDat

Anyone having NHS visits to the dentist?

(1) Yes
(2) No

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

EyeTest

QAccomDat

Anyone having NHS eyetests?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Specs

QAccomDat

Anyone having NHS glasses/lenses?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Hospital

QAccomDat

Anyone having NHS hospital treatment?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

Pres

QAccomDat

Anyone having NHS prescriptions?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

SchMeal

QAccomDat

Anyone having school meals?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

SchMilk

QAccomDat

Anyone having school milk?

(1) Yes
(2) No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

WelfMilk

QAccomDat

Anyone having welfare milk?

(1) Yes
(2) No

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
RESERVECHECK

RESERVECHECK

CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
RESERVECHECK

RESERVECHECK

CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
RESERVECHECK

RESERVECHECK

CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
RESERVECHECK

RESERVECHECK

COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
AND: QAccomdat.HHStat = Shared
AND: In loop FOR Loop1 := 1 TO NewBU

BUHBElig[Loop1] := Yes

COMPUTE ALWAYS:

QAccomdat.AdultH := AllAd

COMPUTE ALWAYS:

QAccomdat.DepChldH := AllCh

COMPUTE ALWAYS:

QAccomdat.DatYrAgo := DLYear

COMPUTE ALWAYS:

QAccomdat.BenUnits := NewBU

FRS0405A.QRenting

Questions about renters

***ASK IF:** QAccomdat.Tenure IN [Part .. Squatting]*

Landlord

QRenting

If property is let through a letting agent or estate agent, the question refers to the owner not the agent, so please probe to try to find out who actually owns the property.

If the respondent does not know who the landlord is, use code 7 (other private individual) rather than coding 'Don't know'.

Code 1 (^LANIHE) includes people renting from Housing Action Trusts.

Code 2 (housing association etc.) includes Registered Social Landlords. Nearly all housing associations are now Registered Social Landlords but continue to be known as housing associations.

(1) ^Council1
(2) A housing association, charitable trust or Local Housing Company
(3) Employer (organisation) of a household member
(4) Another organisation
(5) Relative/friend (before you lived here) of household member
(6) Employer (individual) of a household member
(7) Another individual, private landlord or Letting Agency

***RECORD IF:** QAccomdat.Tenure IN [Part .. Squatting]*

LLEx

QRenting

^KeyTxt

^SuppTxt

OPEN

***WARN IF:** QAccomdat.Tenure IN [Part .. Squatting]*
***AND:** Edit = No*
***AND:** Landlord = NONRESPONSE*
ERROR

^KeyTxt

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Edit = No
AND: Landlord = NONRESPONSE

LLEx

QRenting

^KeyTxt

^SuppTxt

OPEN

COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Landlord = Council

Allowance := 'Rebate'

COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: NOT (Landlord = Council)

Allowance := 'Allowance'

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]

Furnish

QRenting

Is this accomodation provided...

(1) furnished,
(2) partly furnished (eg. curtains and carpets only),
(3) or unfurnished?

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Landlord IN [FrndRel .. OthIndiv]

ResLL

QRenting

Does the landlord live in the building?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = Yes) AND (QAccomdat.TypeAcc = Purpose_built)
```

ResLL2

QRenting

Does the landlord live in the same flat as you or not?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: NatCen <> NI
```

YStart

QRenting

ASK OR CODE
In which year did you first become a tenant of this accommodation?

INTERVIEWER: 'YOU'=PERSON(S) NAMED AT 'Hhldr', THAT IS... ^RentName.

(1) 1988 or earlier
(2) From 1989 to February 1997
(3) March 1997 or later

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: NOT (NatCen <> NI)
```

NIYstart

QRenting

ASK OR CODE
In which year did you first become a tenant of this accommodation?

INTERVIEWER: 'YOU'=PERSON(S) NAMED AT 'Hhldr', THAT IS... ^RentName.

(1) 1978 or earlier
(2) 1979 or later

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: ((NatCen <> NI) AND (YStart IN [ToFeb97 .. AftMar97])) OR ((NatCen
     = NI) AND (NIYstart = Aft1979))
```

Ctract

QRenting

When you started to rent this accommodation ...READ OUT (RUNNING PROMPT)...

(1) ...did you and the landlord sign a written agreement,
(2) ...did you have a written agreement which you didn't sign,
(3) ...or did you just have an unwritten agreement?

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NatCen = NI
```

various := 'various'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NOT (NatCen = NI)
```

various := 'various other'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NatCen <> NI
     AND: (YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 =
     EMPTY OR (ResLL2 = Yes)))
     AND: Scotland = Yes
```

AssuredSH := 'Short Assured'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NatCen <> NI
     AND: (YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 =
     EMPTY OR (ResLL2 = Yes)))
     AND: NOT (Scotland = Yes)
```

AssuredSH := 'Assured Shorthold'

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *Landlord IN [FrndRel .. OthIndiv]*
And: *NatCen <> NI*
And: *(YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 = EMPTY OR (ResLL2 = Yes)))*

TenType

QRenting

SHOW CARD F
Can you tell me what kind of tenancy you have?

INTERVIEWER: IF TENANCY TYPE WRITTEN ON CONTRACT/NOTICE ASK RESPONDENT TO READ OUT

(1) ^AssuredSH
(2) Assured
(3) Regulated (tenancy must have started in 1988 or earlier)
(4) Resident landlord
(5) Let by educational institution
(6) Other type of let

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *Landlord IN [FrndRel .. OthIndiv]*
And: *NatCen <> NI*
And: *(YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 = EMPTY OR (ResLL2 = Yes)))*
And: *TenType = OthLet*

OthType

QRenting

SHOW CARD G
There are ^various ways in which landlords can let accommodation. Will you please look at this card and tell me if the letting is one of these?

(1) Crown tenancy/licence (includes H.M Forces)
(2) Service occupancy (excludes H.M. Forces)
(3) Business or agricultural tenancy
(4) Assured agricultural occupancy
(5) Asylum seeker let (issued by National Asylum Support Service NASS)
(6) Holiday let
(7) Other type of let

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *Landlord IN [FrndRel .. OthIndiv]*
And: *NatCen <> NI*
And: *(YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 = EMPTY OR (ResLL2 = Yes)))*
And: *TenType = AssSHold*

LowShort

QRenting

Is this a low season let?
INTERVIEWER: THIS REFERS TO AN OUT OF SEASON LET

(1) Yes
(2) No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *NOT (NatCen <> NI)*

OthType

QRenting

SHOW CARD G
There are ^various ways in which landlords can let accommodation. Will you please look at this card and tell me if the letting is one of these?

(1) Crown tenancy/licence (includes H.M Forces)
(2) Service occupancy (excludes H.M. Forces)
(3) Business or agricultural tenancy
(4) Assured agricultural occupancy
(5) Asylum seeker let (issued by National Asylum Support Service NASS)
(6) Holiday let
(7) Other type of let

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *Ctract IN [Signed .. NotSign]*
    AND: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
    AND: *NatCen = NI*

Assured := 'a Protected'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *Ctract IN [Signed .. NotSign]*
    AND: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
    AND: *NatCen = NI*

Tenancy := 'a Protected Shorthold Tenancy'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *Ctract IN [Signed .. NotSign]*
    AND: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
    AND: *NatCen = NI*

Shorthold := 'a Protected Shorthold'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *Ctract IN [Signed .. NotSign]*
    AND: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
    AND: *NOT (NatCen = NI)*

Assured := 'an Assured'

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *Landlord IN [FrndRel .. OthIndiv]*
And: *Ctract IN [Signed .. NotSign]*
And: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
And: *NOT (NatCen = NI)*

Tenancy := 'for an Assured Shorthold Tenancy'

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *Landlord IN [FrndRel .. OthIndiv]*
And: *Ctract IN [Signed .. NotSign]*
And: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
And: *NOT (NatCen = NI)*

Shorthold := 'an Assured Shorthold'

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *Landlord IN [FrndRel .. OthIndiv]*
And: *Ctract IN [Signed .. NotSign]*
And: *(((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
And: *NatCen <> NI*
And: *TenType = NONRESPONSE OR OthType = NONRESPONSE*

Short1

QRenting

There is a form of tenancy called ^Assured Shorthold. It had to be initially for a fixed period and you had to be given a special notice in writing by the landlord that told you it was ^Tenancy. Here is an example of a notice to a tenant saying that the tenancy is ^Shorthold.
SHOW EXAMPLE OF NOTICE.

Does your notice state that it is ^Assured Shorthold or not?

(1) Yes, an Assured Shorthold
(2) Other agreement

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = AftMar97)
     AND: TenType = NONRESPONSE OR OthType = NONRESPONSE
```

Short2

QRenting

Most tenancies are Assured Shortholds. There are others, just called 'Assured'. For these you have to be given a notice, in writing by the landlord, that tells you it is NOT an Assured Shorthold agreement.

SHOW EXAMPLE OF NOTICE.

Does your agreement or notice state that it is NOT an Assured Shorthold?

INTERVIEWER: The law changed in March 1997 to the effect that by default all tenancy agreements are assured shortholds, unless the landlord gave written notice to the contrary.

(1) Not an Assured Shorthold
(2) No, other agreement

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR
     (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 =
     No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))
     AND: NatCen = NI
```

assessed := 'assessed'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR
     (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 =
     No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))
     AND: NatCen = NI
```

rent_officer := ('rent officer for Northern Ireland who will set up an independent ' + 'committee to assess and fix an appropriate rent')

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR
     (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 =
     No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))
     AND: NOT (NatCen = NI)
```

assessed := 'registered'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR
     (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 =
     No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))
     AND: NOT (NatCen = NI)
```

rent_officer := ('local rent officer or rent assessment committee to decide on a ' + 'fair rent which is then registered')

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *(((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 = No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))*

FairRent

QRenting

Has the rent been registered by the local rent officer or rent committee?

(1) Yes
(2) No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*

AccJob

QRenting

Does this accommodation go with the present job of anyone in your household?

(1) Yes
(2) No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *AccJob = Yes*

AccJbPer

QRenting

Who is that?

CODE ALL THAT APPLY.

SET [14] OF
(1) ^DMName[1]
(2) ^DMName[2]
(3) ^DMName[3]
(4) ^DMName[4]
(5) ^DMName[5]
(6) ^DMName[6]
(7) ^DMName[7]
(8) ^DMName[8]
(9) ^DMName[9]
(10) ^DMName[10]
(11) ^DMName[11]
(12) ^DMName[12]
(13) ^DMName[13]
(14) ^DMName[14]

CHECK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *AccJob = Yes*
    AND: *In loop FOR Index := 1 TO 14*
    AND: *Index IN AccJbPer*
    PRec[].Depend[Index] = Adult

Code ^Index is not valid for this question.

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *QAccomdat.HHStat = Shared*

es_household := (' you, that is, just ' + HRPNames + ',')

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *QAccomdat.HHStat = Shared*

IsAre := 'Are'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *NOT (QAccomdat.HHStat = Shared)*

es_household := 'es your household'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *NOT (QAccomdat.HHStat = Shared)*

IsAre := 'Is'

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*

RentDoc

QRenting

Do you have a rent book, rent card, Housing Benefit statement or some other rent document that you could consult?

IF HB STATEMENT AVAILABLE PLEASE CONSULT THIS.

(1) Housing Benefit Statement
(2) Some other document
(3) None

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*
AND: *RentDoc IN [HBStmt, Oth]*

Consult_the_document := ' PLEASE CONSULT THE DOCUMENT.'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*
AND: *NOT (RentDoc IN [HBStmt, Oth])*

Consult_the_document := ''

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
```

Rent

QRenting

If in arrears, enter amount last paid but open a Note and give the date of payment (and say if an extra amount is included to pay towards the arrears).
If 100% rent rebate/HB is received and water/sewerage rates and other services etc are not included in the rent, then the amount at 'Rent' should be zero. But if water, etc rates ARE included in the rent, then the amount paid for these rates should be entered at the question 'Rent'.

If rent includes an element for the business part of the property (eg a shop beneath a flat), and the amount for the residential part cannot be determined, enter 'DK' at 'Rent'.

0.00..999997.00

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
```

RentEx

QRenting

^KeyTxt

^SuppTxt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Edit = No
     AND: Rent = NONRESPONSE
     ERROR
```

^KeyTxt

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Edit = No
     AND: Rent = NONRESPONSE
```

RentEx

QRenting

^KeyTxt

^SuppTxt

OPEN

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

RentPx

QRenting

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

RentPd

QRenting

How long does this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd = Note
```

RentPx

QRenting

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: Edit = Yes
     RentPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

RentWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Landlord = Council
     (RentWkly < 101) AND INVOLVING(RentPd,Rent)
```

This comes to ^P^RentWkly a week.
Rents for Council tenants are normally below ^P100 a week.

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Edit = No
     ((RentWkly < 151) OR (Landlord = Council)) AND INVOLVING(RentPd,Rent)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent = REFUSAL
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent = DONTKNOW
```

RentDK

QRenting

INTERVIEWER: IS THIS 'DON'T KNOW' BECAUSE RENT IS PARTLY FOR BUSINESS,
^S4^S10 AND YOU CANNOT ESTABLISH A SEPARATE AMOUNT FOR THE DOMESTIC
^S4^S10 ACCOMMODATION?

(1) Yes (Please give full details in a Note)
(2) No

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent = DONTKNOW
     AND: RentDK <> Yes
```

HMissVar := (HMissVar + 1)

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Rent <> EMPTY*

RentHol

QRenting

Do you have a rent holiday?

INTERVIEWER: SOME PEOPLE KNOW THIS AS 'Rent free week(s)'.

(1) Yes
(2) No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Rent <> EMPTY*
    AND: *RentHol = Yes*

WeekHol

QRenting

For how many weeks of the year do you have a rent holiday?

1..52

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Rent <> EMPTY*
    AND: *RentHol = Yes*
    WeekHol <= 8

Rent holidays do not normally exceed 8 weeks per year.

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord = Council*

allowed := 'allowed'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord = Council*

directly := ''

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *NOT (Landlord = Council)*

allowed := 'receiving'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *NOT (Landlord = Council)*

directly := ', either directly or by having it paid to your landlord'

***ASK IF:** QAccomdat.Tenure IN [Part .. Squatting]*

HBenefit

QRenting

Are you ^allowed Housing Benefit or Rent ^Allowance, to help with paying your rent^directly?

(1) Yes
(2) No

***RECORD IF:** QAccomdat.Tenure IN [Part .. Squatting]*

HBenEx

QRenting

^KeyTxt

^SuppTxt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Edit = No
     AND: HBenefit = NONRESPONSE
     ERROR
```

^KeyTxt

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Edit = No
     AND: HBenefit = NONRESPONSE
```

HBenEx

QRenting

^KeyTxt

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: ((Rent = 0) AND (Rent = RESPONSE)) AND (HBenefit = Yes)
```

Rebate

QRenting

You said that you paid no rent last time, is that because you get 100% Housing Benefit?

(1) Yes
(2) No

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *((Rent = 0) AND (Rent = RESPONSE)) AND ((HBenefit = No) OR (Rebate = No))*

RebateO

QRenting

Can I just check, what is the reason for your paying no rent last time?

STRING[60]

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HHSize > 1*

you_all := 'you and the other members of your household'

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *NOT (HHSize > 1)*

you_all := 'you'

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*

NCDVIB := (NCDVIB + 1)

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*

HBenAmt

QRenting

How much Housing Benefit/ rent rebate/ allowance are ^you_all allowed?

INTERVIEWER: Some respondents may receive more housing benefit than the amount of their rent.

0.01..997.00

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *HBenAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

Record if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *HBenAmt > 0*

HBenPx

QRenting

^Pd97Txt

OPEN

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*

HBenPd

QRenting

How long does this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *HBenPd = Note*

HBenPx

QRenting

^Pd97Txt

OPEN

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *Edit = Yes*
HBenPd <> Note

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[1] := 1
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[2] := 2
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[3] := 3
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[4] := 4
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[5] := 4.333
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[7] := 8.67
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[8] := 6.5
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[9] := 5.78
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0

PdConW[10] := 5.2
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QRenting (continued)

Questions about renters

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *HBenPd IN [OneWeek .. Year]*
AND: *LWeekly1 >= 0.01*

HBenWkly := LWeekly1

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *HBenPd IN [OneWeek .. Year]*
AND: *LWeekly1 >= 0.01*
AND: *Edit = No*
(HBenWkly < 200) AND INVOLVING(HBenPd,HBenAmt)

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *HBenPd IN [OneWeek .. Year]*
AND: *LWeekly1 >= 0.01*
AND: *Edit = Yes*
AND: *(HBenWkly = RentWkly) OR (Rent = HBenAmt)*
ERROR AND INVOLVING(Rent,HBenAmt)

EDITOR: The respondent has given exactly the same figure for rent and housing benefit. Please check that there is no double counting.

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *Rent > 0*

HBenChk

QRenting

Can I just check, is the amount of ^rent for rent that you mentioned earlier, BEFORE or AFTER taking off the Housing Benefit?

(1) Before
(2) After

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt > 0*
AND: *Rent > 0*
AND: *((HBenWkly = RESPONSE) AND (RentWkly = RESPONSE)) AND (HBenChk = Befor)*
(HBenWkly <= RentWkly) AND INVOLVING(HBenAmt,Rent)

Housing Benefit is not normally more than rent. However from October 2003 in some areas Housing Benefit may exceed rent. Please double check the figure with the respondent.

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: (HBenAmt = DONTKNOW) AND ((Rent = DONTKNOW) OR (Rent > 0))
```

RentFull

QRenting

How much is your FULL rent - that is, BEFORE Housing Benefit or Rent ^Allowance?

0.00..999997.00

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: (HBenAmt = DONTKNOW) AND ((Rent = DONTKNOW) OR (Rent > 0))
```

RentPx1

QRenting

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: (HBenAmt = DONTKNOW) AND ((Rent = DONTKNOW) OR (Rent > 0))
```

RentPd1

QRenting

How long does this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *(HBenAmt = DONTKNOW) AND ((Rent = DONTKNOW) OR (Rent > 0))*
AND: *RentPd1 = Note*

RentPx1

QRenting

^Pd97Txt

OPEN

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *(HBenAmt = DONTKNOW) AND ((Rent = DONTKNOW) OR (Rent > 0))*
AND: *Edit = Yes*
RentPd1 <> Note

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*

HBWeeks

QRenting

For how long have you been on Housing Benefit or Rent ^Allowance (this time)?

(1) Up to 2 years
(2) 2 years but less than 3
(3) 3 years but less than 4
(4) 4 years but less than 5
(5) 5 or more years

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBWeeks = Less2Y*

HBWeeks2

QRenting

Please tell me how many weeks you have been on Housing Benefit or Rent ^Allowance (this time)?

0..997

CHECK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBWeeks = Less2Y*
AND: *HBWeeks2 = RESPONSE*
IN(HBWeeks2,[1..104])

Enter a value between 1 and 104

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *RentDoc = HBStmt*

EligAmt

QRenting

On the (rent book/ card/ statement), what is the amount shown for ELIGIBLE RENT?

THIS MUST BE THE ELIGIBLE RENT (MAY NOT BE SAME AS THE AMOUNT OF BENEFIT)
ELIGIBLE RENT = AFTER DEDUCTIONS.

1.00..1000.00

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *RentDoc = HBStmt*
And: *EligAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

Record if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *RentDoc = HBStmt*
And: *EligAmt > 0*

EligPx

QRenting

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

EligPd

QRenting

What period does that cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: EligPd = Note
```

EligPx

QRenting

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: Edit = Yes
     EligPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: EligPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

EligWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: EligPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Edit = No
     (EligWkly < 200) AND INVOLVING(EligPd,EligAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = No
```

HBenWait

QRenting

Are you awaiting the outcome of a claim for Housing Benefit - that is, either Rent Rebate or Rent Allowance?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
```

WSInc

QRenting

Were water or sewerage charges (rates) included in the rent which you mentioned?^Consult_the_document

(1) Both water & sewerage
(2) Water only
(3) Sewerage only
(4) Neither

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSInc = Both
```

COMBINED_AMOUNT := '

INTERVIEWER: ENTER COMBINED AMOUNT.'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSInc = Water
```

water_sewerage := 'water'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSInc = Sewer
```

water_sewerage := 'sewerage'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (WSInc = Sewer)
```

water_sewerage := 'water/sewerage'

FRS0405A.QRenting.PdTxt1()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
```

LInThat := ' in that '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = OneWeek
```

PPdTxt := 'one week period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = TwoWeek
```

PPdTxt := 'two week period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = ThrWeek
```

PPdTxt := 'three week period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = Fourweek
```

PPdTxt := 'four week period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = Month
```

PPdTxt := 'calendar month'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = ThrMonth
```

PPdTxt := 'three month period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = SixMonth
```

PPdTxt := 'six month period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = Year
```

PPdTxt := 'year'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
```

LInThat := ' in those '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = TwoMonth
```

PPdTxt := 'two calendar months'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = EighYear
```

PPdTxt := 'eight payments'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = NineYear
```

PPdTxt := 'nine payments'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = TenYear
```

PPdTxt := 'ten payments'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod = LessWeek
```

LInThat := ' in that '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod = LessWeek
```

PPdTxt := 'week'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: NOT (PPeriod = LessWeek)
```

LInThat := ''

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: NOT (PPeriod = LessWeek)
```

PPdTxt := ''

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
```

PPdTxt := (LInThat + PPdTxt)

FRS0405A.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd IN [OneWeek .. LessWeek])
```

`in_that_period := ''`

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
```

WSIncAmt

QRenting

How much was included for ^water_sewerage ^in_that_period?
^COMBINED_AMOUNT^Consult_the_document

1.00..100.00

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: Rent >= 0
     NOT(WSIncAmt > Rent)
```

The amount INCLUDED in rent for water/sewerage is GREATER than the rent!

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt = RESPONSE) AND (Rent = NONRESPONSE OR RentFull =
     NONRESPONSE)
```

RentPx2

QRenting

^Pd97Txt

OPEN

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *(PTenure IN [Rents, Part]) AND Rent <> EMPTY*
AND: *(Scotland <> Yes) AND (NatCen <> NI)*
AND: *WSInc IN [Both, Water, Sewer]*
AND: *(WSIncAmt = RESPONSE) AND (Rent = NONRESPONSE OR RentFull = NONRESPONSE)*

RentPd2

QRenting

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *(PTenure IN [Rents, Part]) AND Rent <> EMPTY*
AND: *(Scotland <> Yes) AND (NatCen <> NI)*
AND: *WSInc IN [Both, Water, Sewer]*
AND: *(WSIncAmt = RESPONSE) AND (Rent = NONRESPONSE OR RentFull = NONRESPONSE)*
AND: *RentPd2 = Note*

RentPx2

QRenting

^Pd97Txt

OPEN

FRS0405A.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND ((RentPd IN [OneWeek .. Year]) OR (RentPd2 IN
     [OneWeek .. Year]))
     AND: LWeekly1 >= 1
```

WSIWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND ((RentPd IN [OneWeek .. Year]) OR (RentPd2 IN
     [OneWeek .. Year]))
     AND: LWeekly1 >= 1
     AND: Edit = No
     (WSIWkly < 20) AND INVOLVING(WSIncAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSIncAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: Rent <> 0
```

SerInc

QRenting

SHOW CARD M
Does the rent which you mentioned include any of the services shown on this card?^Consult_the_document

CODE ALL THAT APPLY.

SET [5] OF
(1) Heating
(2) Lighting
(3) Hot water
(4) Fuel for cooking
(5) TV licence fees
(6) None of these services

CHECK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *(PTenure IN [Rents, Part]) AND Rent <> EMPTY*
AND: *Rent <> 0*
AND: *None IN SerInc*
SerInc.CARDINAL = 1

'None of these' is an exclusive code for this question.

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *(PTenure IN [Rents, Part]) OR ((PTenure IN [RentFree, Squatting]) AND (AccJob <> Yes))*

AccNonHH

QRenting

(Apart from Housing Benefit) does anyone outside your household pay any rent on this accommodation on your behalf?

INTERVIEWER: EXCLUDE Housing Benefit - ie. RENT REBATE or RENT ALLOWANCE.

(1) Yes
(2) No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*

AccPay

QRenting

Who is that?

CODE ALL THAT APPLY.

SET [5] OF
(1) ^GOVSSA
(2) Employer
(3) Other organisation
(4) Friend or relative
(5) Other

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*
NOT(IN(GOV,AccPay))

Are you sure? ^GOV1 only ever pay arrears of rent. Double-check, that respondent is not thinking of Housing Benefit.
If genuine arrears, suppress this warning.

FRS0405A.QRenting.QAccPay[]

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[1] := GOV1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[2] := 'employer'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[3] := 'other organisation'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[4] := 'relative or friend'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[5] := '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

LRent := PRent

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

RentSeq := PSeq

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

AccPay := PSeq

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

AccAmt

QRenting

How much rent did the ^Payer[AccPay] pay for you last time?

0.01..999997.00

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

AccPx

QRenting

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

AccPd

QRenting

How long did that cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: AccPd = Note
```

AccPx

QRenting

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: Edit = Yes
     AccPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QRenting.QAccPay[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QRenting.QAccPay[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: AccPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

AccWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: AccPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (AccWkly < 151) AND INVOLVING(AccPd,AccAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: PRent > 0
```

AccChk

QRenting

Can I just check, is the amount of ^P^LRent for rent, that you mentioned earlier, BEFORE or AFTER deducting this payment?

(1) Before
(2) After

FRS0405A.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: ((QAccPay[Index].AccWkly = RESPONSE) AND (RentWkly = RESPONSE))
     AND (QAccPay[Index].AccChk = Befor)
```

AccC := (AccC + QAccPay[Index].AccWkly)

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: ((QAccPay[Index].AccWkly = RESPONSE) AND (RentWkly = RESPONSE))
     AND (QAccPay[Index].AccChk = Befor)
     (AccC <= RentWkly) AND
     INVOLVING(QAccPay[Index].AccPd,QAccPay[Index].AccAmt)
```

The amount recorded for help with your rent is greater than the rent recorded.

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Edit = Yes
     AND: Rent = NONRESPONSE OR RentPd = NONRESPONSE
     NOT(IN(Landlord,[???])) AND
     INVOLVING(QAccomdat.Rooms,QAccomdat.TypeAcc)
```

MISSING INFORMATION FOR RENT AMOUNT OR PERIOD.
FOLLOW EDIT INSTRUCTIONS FOR 'RENT'

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Edit = Yes
     AND: HBenAmt = NONRESPONSE OR (HBenPd = NONRESPONSE AND (HBenefit =
     Yes))
     NOT(IN(Landlord,[???]))
```

MISSING INFORMATION FOR Housing Benefit AMOUNT AND/OR PERIOD.
FOLLOW EDIT INSTRUCTIONS FOR 'Housing Benefit'

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: QAccomdat.Tenure = RentFree
     IN(QRenting.Landlord,[???])
```

It is very unusual for Local Authority or Housing Association tenants to be living rent-free. Please check with respondent. Change 'Tenure' to renting if 100% Housing Benefit received, or somebody else pays the rent.

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: Loop1 IN QHholder.HHldr
```

Oldest := (MAX(DMAge[Loop1],Oldest))

FRS0405A.QOwner1

Questions about mortgages

Ask if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*

BuyYear

QOwner1

This should be the year this property was bought. Even if the respondent states that the current mortgage was 'carried over' from a previous property, enter the purchase date for this property - not the previous one.

1901..2005

Check if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
And: *(BuyYear = RESPONSE) AND (POldest > 0)*
BuyYear >= (DLYear.YEAR - POldest)

This is before the date of birth of the oldest householder. Please check your figures.

Check if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
RESERVECHECK

RESERVECHECK

Check if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
RESERVECHECK

RESERVECHECK

Check if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
RESERVECHECK

RESERVECHECK

Check if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *QDataBag.SampMnth IN [4 .. 12]*
BuyYear <> 2005

Wrong Year!

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*

YearLive := ORD(QAccomdat.YearLive)

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *(BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])*
AND: *YearLive = 4*

YearLive := 5

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *(BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])*
AND: *YearLive = 5*

YearLive := 10

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *(BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])*
AND: *YearLive = 6*

YearLive := 20

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *(BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])*

Time[1] := '12 months'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *(BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])*

Time[2] := '2 years'

COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))
AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])

Time[3] := '3 years'

COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))
AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])

Time[5] := '5 years'

COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))
AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])

Time[10] := '10 years'

COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))
AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])

Time[20] := '20 years'

COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))
AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])

MorgYear := (QSignIn.StartDat.YEAR - BuyYear)

WARN IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))
AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
(YearLive >= MorgYear) AND INVOLVING(QAccomdat.YearLive,BuyYear)

The respondent has lived here for less than ^Time[YearLive], but the mortgage started in ^BuyYear - ^MorgYear years ago. Please check that BuyYear is when the mortgage on THIS PROPERTY was taken out. (If so, suppress & continue)

ASK IF: QAccomdat.Tenure IN [Outright .. Part]
AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))

PurcLoan

QOwner1

Can I just check, did you take out one loan to purchase this accommodation, or more than one?

(1) One
(2) Two (or more) loans for purchase

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *PTenure = Part*

your_share_in := ' your share in'

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*

PurcAmt

QOwner1

What was the purchase price of^your_share_in your house/flat?

-99999999.99..999999999.99

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *Edit = No*
PurcAmt < 500000

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *PurcAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*

OthMort3

QOwner1

May I just check, are you currently using this house/flat as security for a mortgage or loan of any kind?

(1) Yes
(2) No

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*

OPur3Rs

QOwner1

This should only apply to loans for purchase. Please resolve, or make a Note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     Paid))
```

OPur3Ex

QOwner1

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     Paid))
     AND: OthMort3 = Yes
```

OthPur3

QOwner1

SHOW CARD N

Which of these items best describe the reasons why you took out this other loan? Any others?

CODE ALL THAT APPLY.

SET [7] OF
(1) To make improvements or extensions to this property
(2) To help purchase a major item like a car, boat, caravan or second home
(3) To get a better, or fixed, interest rate
(4) In connection with a business
(5) To buy out another person's share in the property
(6) For essential repairs to make the property fit for occupation
(7) Some other purpose (INTERVIEWER: SPECIFY IN A NOTE.)

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     Paid))
     AND: OthMort3 = Yes
     NOT(IN(IntrRate,OthPur3))
```

NOT (IntrRate IN OthPur3)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     Paid))
     AND: OthMort3 = Yes
     AND: (OPur3Rs = Suppressed) OR OPur3Ex <> EMPTY
```

OPur3Ex

QOwner1

^SuppTxt

OPEN

FRS0405A.QOwner1.QMortgage.M[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq IN [1 .. 2]
     AND: PPTenure = Part
```

to_buy_this_house := ' to buy your share in this house/flat'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq IN [1 .. 2]
     AND: NOT (PPTenure = Part)
```

to_buy_this_house := ' to buy this house/flat'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq IN [1 .. 2]
```

fill := (', in ' + STR(PBuyYear))

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 3
```

to_buy_this_house := ' for essential repairs'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 1
```

MORTGAGE := (' MAIN MORTGAGE
' + S10 + S4 + 'FOR ' + 'THE PURCHASE OF THIS ACCOMMODATION.')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 1
```

INSTRUC := ('

' + S4 + S4 + '(QUESTIONS ABOUT ANY OTHER, ' + 'SEPARATE MORTGAGE WILL FOLLOW.)

')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 2
```

MORTGAGE := (' SECOND MORTGAGE
' + S10 + S4 + 'FOR ' + 'THE PURCHASE OF THIS ACCOMMODATION.')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 2
```

INSTRUC := ('

' + S4 + S4 + '(QUESTIONS ABOUT ANY OTHER, ' + 'SEPARATE MORTGAGE WILL FOLLOW.)

')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 3
```

MORTGAGE := 'LOAN FOR ESSENTIAL REPAIRS'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
```

IntroM

QOwner1

^S10 THE NEXT QUESTIONS ARE ABOUT THE ^MORTGAGE ^INSTRUC

(1) Press <Enter> to continue.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
```

MortSeq := PSeq

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 2
```

Loan2Y

QOwner1

INTERVIEWER CHECK: DO THEY STILL HAVE THIS OTHER MORTGAGE FOR PURCHASE? (IF NOW REPAID, USE CODE 2)

(1) Yes, still have this mortgage
(2) No, mortgage has been repaid

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
```

LoanYrRs

QOwner1

INTERVIEWER: You have entered that the respondent took out a second loan on this property before they purchased it. Please check your answer. You MUST make a note if you suppress this check.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
```

LoanYrEx

QOwner1

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
```

LoanYear

QOwner1

In which year did you take out this mortgage or loan?

1901..2005

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
     AND: QDataBag.SampMnth IN [4 .. 12]
     LoanYear <> 2005
```

Wrong Year!

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
     AND: (Edit <> Yes) AND ((LoanYear > 0) AND (BuyYear > 0))
     BuyYear <= LoanYear
```

BuyYear <= LoanYear

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
     AND: (Edit <> Yes) AND ((LoanYear > 0) AND (BuyYear > 0))
     AND: (LoanYrRs = Suppressed) OR LoanYrEx <> EMPTY
```

LoanYrEx

QOwner1

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

LPurcAmt := PPurcAmt

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

BorAmtRs

QOwner1

INTERVIEWER: The amount borrowed is more than the purchase price - this is very unusual. Please check your figures and, if necessary, explain in a Note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

BorAmtEx

QOwner1

^SuppTxt

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

BorrAmt

QOwner1

This should be the original amount of this mortgage, as taken out when the property was purchased (in 'BuyYear').

Properties/mortgages partly for business: If the amount borrowed includes the purchase of non-domestic accommodation or land, eg. a farm, a shop with flat above, try to obtain purchase and mortgage details for the domestic element only.

-99999999.99..999999999.99

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((Edit <> Yes) AND (BorrAmt > 0)) AND (LPurcAmt > 0)
     BorrAmt <= LPurcAmt
```

BorrAmt <= LPurcAmt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((Edit <> Yes) AND (BorrAmt > 0)) AND (LPurcAmt > 0)
     AND: (BorAmtRs = Suppressed) OR BorAmtEx <> EMPTY
```

BorAmtEx

QOwner1

^SuppTxt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: BorrAmt = RESPONSE
     BorrAmt < 250000
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: BorrAmt = RESPONSE
     BorrAmt >= 500
```

That seems very low - please check your figures.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (PSeq = 1) AND (BorrAmt = DONTKNOW)
```

BorAmtDK

QOwner1

INTERVIEWER: IS THIS 'DON'T KNOW' BECAUSE THE ORIGINAL MORTGAGE WAS TO BUY DOMESTIC ACCOMMODATION AND FOR BUSINESS PURPOSES, AND YOU CANNOT GET A SEPARATE FIGURE FOR THE DOMESTIC PART?

(1) Yes (Please give full details in a Note)
(2) No

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: BorrAmt = REFUSAL
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((PSeq = 1) AND (BorrAmt = DONTKNOW)) AND (BorAmtDK <> Yes)) OR
     ((PSeq <> 1) AND (BorrAmt = DONTKNOW))
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

RMort

QOwner1

'Re-mortgage': a new mortgage is taken out and is used to REPAY an existing mortgage on a property. Typically this happens when a new lender offers better terms, eg. a lower rate of interest. The new loan can be bigger than the old one.

'Further advance or top-up': the loan is EXTENDED (eg. from ^P30,000 to ^P40,000). But, if there are SEPARATE payments to cover the new sum borrowed, this counts as a separate loan - NOT a further advance/top-up. Separate loans are dealt with later, at 'OthMort1'.

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
```

RMortYr

QOwner1

In which year did you take out the most recent re-mortgage/further advance?

1901..2005

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: QDataBag.SampMnth IN [4 .. 12]
     RMortYr <> 2005
```

Wrong Year!

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: (RMortYr = RESPONSE) AND (PBuyYear = RESPONSE)
     RMortYr >= PBuyYear
```

The first mortgage was taken out in ^PBuyYear, so the re-mortgage can't have been taken out before that. Please amend your answers.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
```

RMAmt

QOwner1

What was the total amount of the mortgage, after re-mortgaging/taking out the further advance?

INTERVIEWER: TOTAL SHOULD BE AFTER ALL RE-MORTGAGES AND FURTHER ADVANCES.

-99999999.99..999999999.99

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: Edit = No
     RMAmt < 250000
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: RMAmt = NONRESPONSE
```

```
HMissVar := (HMissVar + 1)
```

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
```

RMPur

QOwner1

SHOW CARD H
Which of these items best describe the reasons why you took out a re-mortgage/ further advance? Any others? CODE ALL THAT APPLY.

SET [8] OF
(1) To make improvements or extensions to this property
(2) To help purchase a major item like a car, boat, caravan or second home
(3) To get a better, or fixed, interest rate
(4) In connection with a business
(5) To buy out another person's share in the property
(6) For essential repairs to make the property fit for occupation
(7) To move to a more flexible mortgage
(8) Some other purpose (SPECIFY IN A NOTE.)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

MortTyEx

QOwner1

^MortTTxt

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

MortType

QOwner1

SHOW CARD I
Looking at this card, please tell me which of these options describe your mortgage or loan?
INTERVIEWER: If necessary add 'With a repayment mortgage, by repaying the original loan we mean the original capital sum borrowed.'

(1) an ENDOWMENT mortgage
(where your mortgage payments cover interest only)
(2) a REPAYMENT mortgage
(where your mortgage payments cover interest and part of the original loan)
(3) a PENSION mortgage
(where your mortgage payments cover interest only)
(4) a PEP, Unit Trust or ISA mortgage
(5) both an endowment (or other interest only) AND a repayment mortgage
(6) an interest only mortgage with more than one linked investment
(e.g. pension and unit trust, endowment and ISA)
(7) an interest only mortgage with NO linked investment
(e.g. NO endowment, pension, PEP or ISA)
(8) or another type (not listed above)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = NONRESPONSE
```

MortTTxt := KeyTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = NONRESPONSE
     ERROR
```

^MortTTxt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = NONRESPONSE
```

MortTyEx

QOwner1

^MortTTxt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = Other
```

MortTTxt := ('INTERVIEWER: This is a rather unusual sort of ' + 'mortgage. Are you sure that it is not covered ' + 'by one of the codes above? If it is not, please ' + 'record the title in a note and answer follow-up ' + 'questions as fully as possible.')

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = Other
     ERROR
```

^MortTTxt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = Other
```

MortTyEx

QOwner1

^MortTTxt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = Endow
```

this_kind_of := 'an endowment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = Repay
```

this_kind_of := 'a repayment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = Pension
```

this_kind_of := 'a pension'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = PEP
```

this_kind_of := 'a PEP, Unit Trust or ISA'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = EndRep
```

this_kind_of := 'an endowment & repayment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [IntLink, IntNoLnk]
```

this_kind_of := 'an interest only'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (MortType IN [IntLink, IntNoLnk])
```

this_kind_of := 'this kind of'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MortType IN [Endow, EndRep]
```

Is_the := 'Apart from the endowment I mentioned earlier is the'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: NOT (MortType IN [Endow, EndRep])
```

Is_the := 'Is the'

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR i := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *MortType IN [Endow, Pension .. Other]*

EndwPrin

QOwner1

SHOW CARD J
^Is_the repayment of the original loan covered by any of the things on this card?
CODE ALL THAT APPLY.

SET [4] OF
(1) Current payments into a Pension Plan (pension mortgage)
(2) Current payments into a PEP or ISA
(3) Current payments into a Unit Trust or Investment Trust scheme
(4) Current payments into any other savings/investment scheme
(5) Proceeds of sale from existing house only
(6) None of the above.

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR i := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *MortType IN [Endow, Pension .. Other]*
    AND: *None IN EndwPrin*
    EndwPrin.CARDINAL = 1

None is an exclusive code for this question.

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR i := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *MortType IN [Endow, Pension .. Other]*
    RESERVECHECK

RESERVECHECK

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

EndwP1Ex

QOwner1

Please leave a note to describe how the respondent will be repaying their mortgage.

^SuppTxt

OPEN

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

EndwP2Ex

QOwner1

^EndP2Txt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

EndwPTxt := ('This method of capital repayment does not match the ' + 'type of mortgage recorded earlier at MortType. ' + ' Please resolve, or make a note.')

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: None IN EndwPrin
     AND: NOT (MortType IN [Endow, EndRep])
     ERROR AND INVOLVING(MortType,EndwPrin)
```

Please leave a note to describe how the respondent will be repaying their mortgage.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: None IN EndwPrin
     AND: NOT (MortType IN [Endow, EndRep])
```

EndwP1Ex

QOwner1

Please leave a note to describe how the respondent will be repaying their mortgage.

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Endow
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndP2Txt := EndwPTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Endow
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
     ERROR AND INVOLVING(MortType,EndwPrin)
```

^EndP2Txt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Endow
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndwP2Ex

QOwner1

^EndP2Txt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Pension
     AND: (((PEP IN EndwPrin) OR (UnitT IN EndwPrin)) OR (OthSch IN
     EndwPrin)) OR (None IN EndwPrin)
```

EndP2Txt := EndwPTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Pension
     AND: (((PEP IN EndwPrin) OR (UnitT IN EndwPrin)) OR (OthSch IN
     EndwPrin)) OR (None IN EndwPrin)
     ERROR AND INVOLVING(MortType,EndwPrin)
```

^EndP2Txt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Pension
     AND: (((PEP IN EndwPrin) OR (UnitT IN EndwPrin)) OR (OthSch IN
     EndwPrin)) OR (None IN EndwPrin)
```

EndwP2Ex

QOwner1

^EndP2Txt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = PEP
     AND: ((Pension IN EndwPrin) OR (OthSch IN EndwPrin)) OR (None IN
     EndwPrin)
```

EndP2Txt := EndwPTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = PEP
     AND: ((Pension IN EndwPrin) OR (OthSch IN EndwPrin)) OR (None IN
     EndwPrin)
     ERROR AND INVOLVING(MortType,EndwPrin)
```

^EndP2Txt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = PEP
     AND: ((Pension IN EndwPrin) OR (OthSch IN EndwPrin)) OR (None IN
     EndwPrin)
```

EndwP2Ex

QOwner1

^EndP2Txt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = IntNoLnk
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndP2Txt := ('You described your mortgage as an interest ' + 'only with NO linked investments, can I just ' + 'check is this savings/investment scheme linked ' + 'to your mortgage? Please amend the answer at ' + 'MortType as appropriate.')

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = IntNoLnk
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
     ERROR AND INVOLVING(EndwPrin,MortType)
```

^EndP2Txt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = IntNoLnk
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndwP2Ex

QOwner1

^EndP2Txt

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Other
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

```
EndP2Txt := ('If there is a pension, ISA or other ' +
'savings/investment scheme to cover the ' + 'repayment of the
original loan, please ' + 'amend the answer at MortType as
appropriate.')
```

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Other
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
     ERROR AND INVOLVING(EndwPrin,MortType)
```

^EndP2Txt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Other
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndwP2Ex

QOwner1

^EndP2Txt

^SuppTxt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Pension IN EndwPrin
```

DMAEndwPrin := Pension

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

payment1 := 'contribution to the'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

payment2 := '(pension plan/PEP/ISA/Unit Trust)'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = IntLink
```

payment1 := 'premium/payment on the'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = IntLink
```

payment2 := ('endowment policy / pension / unit trust /' + 'ISA / PEP / investment trust / other policy')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (MortType = IntLink)
```

payment1 := 'premium on the'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT (MortType = IntLink)*

payment2 := 'endowment policy'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *MortType = IntLink*

policy := ('policy / pension / unit trust / ISA / PEP / investment ' + 'trust / other policy')

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT (MortType = IntLink)*

policy := 'policy'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *MortType IN [Endow, Pension .. Other]*
AND: *MortType IN [Endow, EndRep]*

Are := '(Can I just check), are'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *MortType IN [Endow, Pension .. Other]*
AND: *NOT (MortType IN [Endow, EndRep])*

Are := 'Are'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

MenPol

QOwner1

^Are there any endowment policies covering the repayment of this mortgage or loan?

(1) Yes
(2) No

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType IN [Endow, EndRep]
     ((MenPol = Yes) OR (MortType = Other)) AND INVOLVING(MortType,MenPol)
```

Normally there would be an endowment policy, with an ENDOWMENT mortgage: please check.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MenPol = Yes
     NOT(IN(MortType,[???,???])) AND INVOLVING(MortType,MenPol)
```

You described your mortgage as an interest only with NO linked investments or another type of mortgage, can I just check is this savings/investment scheme linked to your mortgage? Please amend the answer at MortType as appropriate.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MenPol = Yes
     (MortType <> Other) AND INVOLVING(MortType,MenPol)
```

If there is an endowment, pension, ISA or other arrangement to cover the repayment of the original loan, please amend the answer at MortType as appropriate.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MenPol = No
     (MortType <> Other) AND INVOLVING(MortType,MenPol)
```

It is unusual for there to be no policies to cover the repayment of the loan. Please check. If original loan is included in monthly payments, please amend your answer at MortType to a Repayment (code 2) OR Endowment and Repayment (code 5) mortgage.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = No
```

How_Long := ('How long is the term of your mortgage. By this we mean ' + 'the agreed term?')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (RMort = No)
```

How_Long := ('How long is the term of your mortgage. By this we ' + 'mean the agreed term since you have remortgaged or ' + 'extended the original loan?')

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

MorFlc

QOwner1

All-in-one accounts are a new type of flexible mortgage which allow a person to link together accounts - for example, a current account, a savings account and a mortgage (as well as other types of loans). There are two types of all-in-one account: current account mortgages and offset mortgages.

INTERVIEWER: Examples include a Virgin-One account, a Woolwich Open Plan or some other all-in-one account.

Is your mortgage an all-in-one account?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MorFlc = Yes
```

MorAll

QOwner1

SHOW CARDS K AND L.

Is your all-in-one account mortgage, a current account mortgage or an offset mortgage?

(1) Current account mortgage
(2) Offset mortgage

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

MortEnd

QOwner1

^How_Long

INTERVIEWER: IF REMORTGAGED AGREED TERM IS FROM THE POINT OF REMORTGAGING.

1..60

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = No
```

remortgage := 'mortgage was taken out'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (RMort = No)
```

remortgage := 'last re-mortgage'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     MortEnd <= 40
```

Are you sure? The end-date would not normally be more than 40 years after the ^remortgage. Please check your figures.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *MorAll = Current*

What_amount := ('What is the amount of the negative balance or ' + 'overdraft on your current account mortgage')

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT (MorAll = Current)*

What_amount := ('What is the amount still outstanding on your ' + 'mortgage/loan from this source - that is, how ' + 'much do you still have to pay off')

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

MortL1Rs

QOwner1

For ^this_kind_of mortgage, the amount outstanding should equal the ^amount ^borrowed. Please check and amend, else explain in a Note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

MortL1Ex

QOwner1

^SuppTxt

OPEN

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

MortL2Rs

QOwner1

For ^this_kind_of mortgage, the amount outstanding should be less than the ^amount ^borrowed. Please check and amend, else explain in a Note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

MortL2Ex

QOwner1

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

MortLeft

QOwner1

^What_amount?

-99999999.99..999999999.99

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortLeft = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
```

borrowed := 'of the re-mortgage'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
```

amount := 'total amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
     (ABS(MortLeft - RMAmt) <= 50) AND INVOLVING(MortLeft)
```

(ABS (MortLeft - RMAmt) <= 50) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
     AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

MortL1Ex

QOwner1

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
```

borrowed := 'originally borrowed'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
     (ABS(MortLeft - BorrAmt) <= 50) AND INVOLVING(MortLeft)
```

(ABS (MortLeft - BorrAmt) <= 50) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
     AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

MortL1Ex

QOwner1

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: RMort = Yes
```

borrowed := 'of re-mortgage'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: RMort = Yes
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: RMort = Yes
     (MortLeft < RMAmt) AND INVOLVING(MortLeft)
```

(MortLeft < RMAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: RMort = Yes
     AND: (MortL2Rs = Suppressed) OR MortL2Ex <> EMPTY
```

MortL2Ex

QOwner1

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
```

borrowed := 'originally borrowed'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
     (MortLeft < BorrAmt) AND INVOLVING(MortLeft)
```

(MortLeft < BorrAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
     AND: (MortL2Rs = Suppressed) OR MortL2Ex <> EMPTY
```

MortL2Ex

QOwner1

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
```

borrowed := 'of the re-mortgage'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
     (MortLeft = RMAmt) AND INVOLVING(MortLeft)
```

(MortLeft = RMAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
     AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

MortL1Ex

QOwner1

^SuppTxt

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
```

borrowed := 'originally borrowed'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
     (MortLeft = BorrAmt) AND INVOLVING(MortLeft)
```

(MortLeft = BorrAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
     AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

MortL1Ex

QOwner1

^SuppTxt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
```

taking_out_the_loan := 'you re-mortgaged'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT (RMort = Yes)
```

taking_out_the_loan := 'taking out the original loan'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
```

MorInPay

QOwner1

How much was your last payment on this mortgage or loan?

0.00..9999.97

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPay = RESPONSE
```

LastPay := STR(MorInPay,6,2)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPay = DONTKNOW
```

LastPay := '??????'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPay = DONTKNOW
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPay = REFUSAL
```

LastPay := '!!!!!!'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPay = REFUSAL
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
```

MorInPx

QOwner1

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
```

MorInPd

QOwner1

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd = Note
```

MorInPx

QOwner1

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: Edit = Yes
     MorInPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
```

MorInUs

QOwner1

Is this the amount you usually pay each time?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
```

MorUs

QOwner1

If the last interest payment includes arrears accept the actual amount given but if it was a long time ago make a note using <Ctrl+M> to give the date, or if the amount was greater than normal to cover past arrears. Payments by people outside the household should be included.

0.00..9999.97

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
```

MorUPx

QOwner1

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
```

MorUPd

QOwner1

How long does this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
     AND: MorUPd = Note
```

MorUPx

QOwner1

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
     AND: Edit = Yes
     MorUPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

MorIWkly := LWeekly

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
```

EPIntC := (((MorIWkly * 52) / MortLeft) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
```

IntFill := ROUND(EPIntC)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     AND: EPIntC <= 2
```

higher := 'lower'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     AND: EPIntC >= 11
```

higher := 'higher'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     ((EPIntC > 2) AND (EPIntC < 11)) AND INVOLVING(MorInPd,MorInPay)
```

The interest payments work out roughly at ^IntFill per cent which is ^higher than most current interest rates available for a mortgage of this size.
If no particular reason for this, please check your answers.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
  AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  AND: *In loop FOR i := 1 TO 3*
  AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  AND: *Loan2Y <> Repaid*
  RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
  AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  AND: *In loop FOR i := 1 TO 3*
  AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  AND: *Loan2Y <> Repaid*
  RESERVECHECK

RESERVECHECK

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
  AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  AND: *In loop FOR i := 1 TO 3*
  AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  AND: *Loan2Y <> Repaid*

MenPolAm0 := Yes

FRS0405A.QOwner1.QMortgage.M[].QEndow[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

MortSeq := PPSeq

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

EndowSeq := PCount

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
     AND: PCount > 1
```

next := 'next'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
     AND: NOT (PCount > 1)
```

next := 'first'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
```

premium_payment := 'premium'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
```

policies_plans := 'endowment policies'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: NOT (PMenpol = Yes)
```

premium_payment := 'payment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: NOT (PMenpol = Yes)
```

policies_plans := 'savings or investment plans'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

MenPolAm

QOwner1

How much was your last ^payment1 ^next ^payment2?
INTERVIEWER: FOR INTEREST ONLY MORTGAGES INCLUDE COMBINED INTEREST AND ENDOWMENT PAYMENT.

0.00..9999.97

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

MenPolPx

QOwner1

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

MenPolPd

QOwner1

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: MenPolPd = Note
```

MenPolPx

QOwner1

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: Edit = Yes
     MenPolPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QOwner1.QMortgage.M[].QEndow[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[3] := 3

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*

PdConW[4] := 4

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*

PdConW[5] := 4.333

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*

PdConW[7] := 8.67

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[].QEndow[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: MenPolPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

MenPWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: MenPolPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: Edit = No
     (MenPWkly < 100) AND INVOLVING(MenPolPd,MenPolAm)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

***ASK IF:** QAccomdat.Tenure IN [Outright .. Part]*
    ***AND:** PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    ***AND:** In loop FOR i := 1 TO 3*
    ***AND:** (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    ***AND:** Loan2Y <> Repaid*
    ***AND:** ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
    ***AND:** MorAll <> Current*
    ***AND:** In loop FOR Count := 1 TO 4*
    ***AND:** (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
    ***AND:** (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)*

IncInInt

QOwner1

Is this ^premium_payment included in the amount you mentioned earlier (^P^PLastPay)?

(1) Yes
(2) No

***RECORD IF:** QAccomdat.Tenure IN [Outright .. Part]*
    ***AND:** PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    ***AND:** In loop FOR i := 1 TO 3*
    ***AND:** (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    ***AND:** Loan2Y <> Repaid*
    ***AND:** ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
    ***AND:** MorAll <> Current*
    ***AND:** In loop FOR Count := 1 TO 4*
    ***AND:** (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
    ***AND:** (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)*
    ***AND:** PMenpol = Yes*

MenstRs

QOwner1

Are you sure? That means the endowment was purchased before HHldr was 18.
INTERVIEWER: Check the circumstances and explain in a note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
```

MenstEx

QOwner1

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
```

MenstYr

QOwner1

In what year was this endowment ^policy taken out?

1901..2005

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
     AND: QDataBag.SampMnth IN [4 .. 12]
     MenstYr <> 2005
```

Wrong Year!

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
     AND: (((Edit <> Yes) AND (QSignIn.StartDat = RESPONSE)) AND
     (HHG.P[QHholder.DVHRPNum].AgeOf > 0)) AND (MenstYr > 0)
     (MenstYr >= ((QSignIn.StartDat.YEAR -
     HHG.P[].AgeOf[QHholder.DVHRPNum]) + 17)) AND INVOLVING(MenstYr)
```

(MenstYr >= ((QSignIn.StartDat.YEAR - HHG.P[QHholder.DVHRPNum].AgeOf) + 17)) AND INVOLVING (MenstYr)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
     AND: (((Edit <> Yes) AND (QSignIn.StartDat = RESPONSE)) AND
     (HHG.P[QHholder.DVHRPNum].AgeOf > 0)) AND (MenstYr > 0)
     AND: (MenstRs = Suppressed) OR MenstEx <> EMPTY
```

MenstEx

QOwner1

^SuppTxt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PCount < 4
```

MpMore

QOwner1

Can I just check, are there any more savings or investment plans covering the repayment of the mortgage or loan?

(1) Yes
(2) No

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
RESERVECHECK

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QOwner1.QMortgage.M[] (continued)

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     (QEndow[Count].MenPWkly <= MorIWkly) AND
     INVOLVING(QEndow[Count].MenPolAm)
```

The payment/endowment premium is more than the last mortgage payment at MorInPay. This is very unusual - please check your figures.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (MorIWkly > 0) AND (QEndow[Count].IncInInt = Yes)
     (QEndow[Count].MenPWkly < MorIWkly) AND
     INVOLVING(QEndow[Count].MenPolAm,QEndow[Count].MenPolPd)
```

The payment/endowment premium is included in the interest payment of ^P^LastPay, so it can't exceed this amount. Please check your figures.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: QEndow[Count].MenPolAm <> 0
```

MenPolAm0 := No

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: (MortSeq = 1) AND (QEndow[1].MpMore = No)
     NOT(QMortgage.M[].MortType[MortSeq] = IntLink) AND
     INVOLVING(QEndow[1].MpMore)
```

Are you sure? Earlier the respondent said they had more than 1 savings / investment policy for this mortgage. Check if the respondent has another policy to tell you about. If not you MUST make a note of the circumstances.

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
```

MpMore

QOwner1

Are there any more policies/plans covering the repayment of the mortgage or loan?

(1) Yes
(2) No

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: QEndow[1].MpMore = Yes
```

MpMore := Yes

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
```

IntPrPay

QOwner1

How much was your last payment on this mortgage or loan?

0.00..9999.97

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: (Edit = No) AND (IntPrPay = RESPONSE)
     NOT(IntPrPay = 0)
```

You have entered that the respondent's last instalment on the mortgage/loan was ^P0. Please do not enter zero even if they paid nothing last time. Please collect the amount they usually pay or if there is no usual, the contractual or notional amount they would need to pay in order for the mortgage/loan to be paid off in the agreed period.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = RESPONSE
```

LastPay := STR(IntPrPay,6,2)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = DONTKNOW
```

LastPay := '??????'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = DONTKNOW
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = REFUSAL
```

LastPay := '!!!!!!'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = REFUSAL
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
```

IntPrPx

QOwner1

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
```

IntPrPd

QOwner1

How long did this cover?

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPd = Note
```

IntPrPx

QOwner1

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: Edit = Yes
     IntPrPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IntPWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     (IntPWkly < 650) AND INVOLVING(IntPrPd,IntPrPay)
```

Are you sure? That is higher than the amount usually entered here. Confirm that the last payment was this amount and if Yes suppress check.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: RMAmt > 0
```

PrIntC := ((((IntPWkly - (RMAmt / (25 * 52))) * 52) / (RMAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: BorrAmt > 0
```

PrIntC := (((IntPWkly - (BorrAmt / (25 * 52))) * 52) / (BorrAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC < 3
```

higher := 'lower'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC > 10
```

higher := 'higher'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     ((PrIntC >= 3) AND (PrIntC <= 10)) AND INVOLVING(IntPrPay)
```

You have entered an amount that is ^higher than that usually paid for a mortgage of this size. Please check that you have entered the correct payment.

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*

IntrUs

QOwner1

Is this the amount you usually pay each time?

(1) Yes
(2) No

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*

IntrU

QOwner1

How much are your usual payments on this mortgage or loan?

0.00..9999.97

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*

IntrPx

QOwner1

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
```

IntrPd

QOwner1

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: IntrPd = Note
```

IntrPx

QOwner1

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: Edit = Yes
     IntrPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IntPWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     (IntPWkly < 650) AND INVOLVING(IntrPd,IntrU)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: RMAmt > 0
```

PrIntC := ((((IntPWkly - (RMAmt / (25 * 52))) * 52) / (RMAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: BorrAmt > 0
```

PrIntC := ((((IntPWkly - (BorrAmt / (25 * 52))) * 52) / (BorrAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC < 3
```

higher := 'lower'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC > 10
```

higher := 'higher'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     ((PrIntC >= 3) AND (PrIntC <= 10)) AND INVOLVING(IntrU)
```

You have entered an amount that is ^higher than that usually paid for a mortgage of this size. Please check that you have entered the correct payment.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MenPol = Yes
```

Apart_do := ('Apart from any endowment policies already ' + 'mentioned, do')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MenPol = Yes
```

redundancy := ' or redundancy'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MenPol = Yes
```

death := '(NOT USED)'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: NOT (MenPol = Yes)
```

Apart_do := 'Do'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: NOT (MenPol = Yes)
```

redundancy := ', redundancy or death'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: NOT (MenPol = Yes)
```

death := 'Death'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
```

MortProt

QOwner1

Only include policies which specifically pay the mortgage. Do not confuse these with polices that simply pay out money in the event of redundancy or sickness (and could be used to pay for anything).

(1) Yes
(2) No

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: (MortProt = RESPONSE) AND (MortType <> Endow)
     MortProt = Yes
```

INTERVIEWER: for this type of mortgage there is normally a protection policy. Please check - is it included in the last mortgage payment? (If no policy, suppress warning and continue.)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MortProt = Yes*

MPCover

QOwner1

What is covered by the mortgage protection policy?

PROBE TO CLASSIFY.
CODE ALL THAT APPLY.

SET [3] OF
(1) Sickness/accident
(2) Redundancy/loss of employment
(3) ^death

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MortProt = Yes*
AND: *MenPol = Yes*
NOT(IN(Dead,MPCover))

This code is not valid for this question.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: MPCover.CARDINAL > 1
```

MPolNo

QOwner1

Can I check, is there one mortgage protection policy, or more than one?

INTERVIEWER: COUNT AS SEPARATE POLICY IF SEPARATE PAYMENTS (PREMIUMS) ARE MADE.

ENTER NUMBER OF POLICIES.

1..3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: IntPrPay <> EMPTY OR (MorInPay > 0)
```

PCP := ('your last payment on the mortgage/loan (' + P + LastPay + ')')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: IntPrPay <> EMPTY OR (MorInPay > 0)
     AND: MorInPay > 0
     AND: MenPolAm0 = No
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := (PCP + ' or in the (pension/PEP/ISA/Unit' + ' Trust) contribution')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: IntPrPay <> EMPTY OR (MorInPay > 0)
     AND: MorInPay > 0
     AND: MenPolAm0 = No
     AND: NOT (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := (PCP + ' or in the endowment premium')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: MorInPay <> EMPTY AND (MenPolAm0 = No)
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := 'the (pension/PEP/ISA/Unit Trust) contribution'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: MorInPay <> EMPTY AND (MenPolAm0 = No)
     AND: NOT (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := 'the endowment premium'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PC := 'pension/PEP/ISA/Unit Trust contribution'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: NOT ((((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PC := 'endowment premium'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

Order[1] := 'FIRST'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

Order[2] := 'SECOND'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

Order[3] := 'THIRD'

FRS0405A.QOwner1.QMortgage.M[].QMortProt[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
```

LPayment_etc := ppayment

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: LPayment_etc =
```

LPayment_etc := 'the mortgage payment you mentioned earlier'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
```

IncMPAmt

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

@?If the precise amount for the mortgage protection policy cannot be given, please ask the respondent to given an estimate rather than accept DK.

0.00..9997.99

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

IncMPPx

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

IncMPPd

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: IncMPPd = Note
```

IncMPPx

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: Edit = Yes
     IncMPPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QOwner1.QMortgage.M[].QMortProt[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[].QMortProt[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: IncMPPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IncMWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: IncMPPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     (IncMWkly < 30) AND INVOLVING(IncMPPd,IncMPAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
```

IncMStYr

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

In what year was the mortgage protection policy taken out?

1901..2005

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: QDataBag.SampMnth IN [4 .. 12]
     IncMStYr <> 2005
```

Wrong Year!

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: (IncMPAmt > 0) OR IncMPAmt = NONRESPONSE
```

IncMP

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

Was this mortgage protection payment included in ^LPayment_etc?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: (IncMPAmt > 0) OR IncMPAmt = NONRESPONSE
     AND: ((IncMP = Yes) AND (SUBSTRING (PLastPay, 1, 1) <> 0)) AND
     (PMenPolAm0 = No)
```

IncMIncl

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

INTERVIEWER: ASK OR CODE.
Was it included in the mortgage payment or the ^ppremium?

(1) mortgage payment
(2) ^ppremium

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QOwner1.QMortgage.M[] (continued)

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: (PSeq IN [1 .. 2]) AND (QMortProt[Count].IncMStYr = RESPONSE)
     PBuyYear <= QMortProt[Count].IncMStYr
```

The mortgage protection policy was taken out BEFORE the mortgage started ('BuyYear'). This seems very unusual - please check your dates.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IntPWkly = RESPONSE
     (QMortProt[Count].IncMWkly <= IntPWkly) AND
     INVOLVING(QMortProt[Count].IncMPAmt,QMortProt[Count].IncMPPd,IntPrPay)
```

The mortgage protection premium is more than the last mortgage payment at IntPrPay (^P^LastPay). This is very unusual - please check your figures.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: MorAll = Current
     QMortProt[Count].IncMPAmt <> DONTKNOW
```

Interviewer: Please try not to collect Don't Know at this question. Ask your respondent to be more specific and provide you with their best estimate as to their monthly payment.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: Edit = Yes
     QMortProt[Count].IncMPAmt<>NONRESPONSE
```

MISSING AMOUNT FOR Mortgage Protection Policy. NOTE THE SIZE OF LAST MORTGAGE PAYMENT (^P^LastPay), THEN FOLLOW Edit Instructions TO FILL IN IncMPAmt.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

OutsMort

QOwner1

Does anyone from outside the household pay anything towards THIS mortgage/loan on your behalf, on a regular basis?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

QOutsPay

QOwner1

Who is that?

SET [6] OF
(1) ^GOV2
(2) Employer
(3) Other organisation
(4) Friend or relative
(5) Mortgage protection/insurance policy
(6) Other

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

Payer[1] := GOV1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

Payer[2] := 'employer'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

Payer[3] := 'other organisation'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

Payer[4] := 'relative or friend'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

Payer[5] := 'policy'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
```

Payer[6] := '

FRS0405A.QOwner1.QMortgage.M[].QOutside[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

MortSeq := PPSeq

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

ContSeq := POutsPay

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

OutsPay := POutsPay

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

OutsAmt

QOwner1

How much did the ^PPayer pay last time?

0.01..999997.00

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

OutsPx

QOwner1

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

OutsPd

QOwner1

How long did that cover?

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: OutsPd = Note
```

OutsPx

QOwner1

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: Edit = Yes
     OutsPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QOwner1.QMortgage.M[].QOutside[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QOwner1.QMortgage.M[].QOutside[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: OutsPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

OutWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: OutsPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (OutWkly < 159) AND INVOLVING(OutsPd,OutsAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: (OutsAmt <> 0) AND (SUBSTRING (PLastPay, 1, 1) <> 0)
```

OutsIncl

QOwner1

Was this included in the mortgage payment that you mentioned earlier?

(1) Yes
(2) No

FRS0405A.QOwner1.QMortgage.M[] (continued)

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*
AND: *In loop FOR Count := 1 TO 6*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*
AND: *In loop FOR Count := 1 TO 6*
RESERVECHECK

RESERVECHECK

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(PSeq = 1) AND (PBuyYear > 1980)*

ExRent

QOwner1

Had you been renting this house/flat before deciding to buy it?

'YOU' = HRP/HOUSEHOLDER, OR SPOUSE/PARTNER

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
    AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
    AND: In loop FOR i := 1 TO 3
    AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
    AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
    IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
    AND: Loan2Y <> Repaid
    AND: (PSeq = 1) AND (PBuyYear > 1980)
    AND: ExRent = Yes
```

RentFrom

QOwner1

Who was it rented from?

PROMPT AS NECESSARY.

(1) ^Council2
(2) Housing Association, co-operative, charitable trust
(3) Employer
(4) Other organisation
(5) Other individual

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
    AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
    AND: In loop FOR i := 1 TO 3
    AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
    AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
    IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
    AND: (PSeq = 1) AND (PPurcLoan = One)
```

OthMort1

QOwner1

I have already asked you about the loan you had to purchase this house/flat. Apart from that, do you have any OTHER mortgage or loan secured on this property?

(1) Yes
(2) No

Ask if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
And: *In loop FOR i := 1 TO 3*
And: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
And: *(PSeq = 2) AND (PPurcLoan = Two)*

OthMort2

QOwner1

May I just check, are you currently using this house/flat as security for a mortgage or loan of any other kind?

(1) Yes
(2) No

Record if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
And: *In loop FOR i := 1 TO 3*
And: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
And: *(OthMort1 = Yes) OR (OthMort2 = Yes)*

OthPurRs

QOwner1

This should only apply to loans for purchase. Please resolve, or make a Note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

Record if: *QAccomdat.Tenure IN [Outright .. Part]*
And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
And: *In loop FOR i := 1 TO 3*
And: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
And: *(OthMort1 = Yes) OR (OthMort2 = Yes)*

OthPurEx

QOwner1

^SuppTxt

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *(OthMort1 = Yes) OR (OthMort2 = Yes)*

OthPur

QOwner1

SHOW CARD N

Which of these items best describe the reasons why you took out the other loan or loans? Any others? CODE ALL THAT APPLY.

SET [7] OF
(1) To make improvements or extensions to this property
(2) To help purchase a major item like a car, boat, caravan or second home
(3) To get a better, or fixed, interest rate
(4) In connection with a business
(5) To buy out another person's share in the property
(6) For essential repairs to make the property fit for occupation
(7) Some other purpose (INTERVIEWER: SPECIFY IN A NOTE.)

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *(OthMort1 = Yes) OR (OthMort2 = Yes)*
NOT(IN(IntrRate,OthPur))

NOT (IntrRate IN OthPur)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *(OthMort1 = Yes) OR (OthMort2 = Yes)*
AND: *(OthPurRs = Suppressed) OR OthPurEx <> EMPTY*

OthPurEx

QOwner1

^SuppTxt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     AND: NOT (MortType IN [Endow, EndRep])
     NOT(IN(None,EndwPrin))
```

EDITOR: MORTGAGE CAPITAL REPAID BY 'UNKNOWN' METHOD: THERE SHOULD BE A NOTE ATTACHED. PLEASE RE-CODE INTO 1-4, IF POSSIBLE.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     IntPrPay<>NONRESPONSE AND IntPrPd<>NONRESPONSE
```

MISSING AMOUNT AND/OR PERIOD FOR Mortgage Instalment.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     MorInPay<>NONRESPONSE AND MorInPd<>NONRESPONSE
```

MISSING AMOUNT AND/OR PERIOD FOR Mortgage Instalment.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     MenPol <> No
```

THERE ARE NO ENDOWMENT POLICIES COVERING THE REPAYMENT OF THIS MORTGAGE OR LOAN.

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two)
     AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs
     IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: (RMAmt = RESPONSE) AND (BorrAmt = RESPONSE)
     BorrAmt <= RMAmt
```

The re-mortgage amount would normally be at least as large as the original mortgage.
Please check your figures.

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 2))) OR ((i = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

FRS0405A.QOwner1.QMortgage

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR i := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QOwner1 (continued)

Questions about mortgages

WARN IF: *`QAccomdat.Tenure IN [Outright .. Part]`*
RESERVECHECK

RESERVECHECK

WARN IF: *`QAccomdat.Tenure IN [Outright .. Part]`*
RESERVECHECK

RESERVECHECK

WARN IF: *`QAccomdat.Tenure IN [Outright .. Part]`*
RESERVECHECK

RESERVECHECK

WARN IF: *`QAccomdat.Tenure IN [Outright .. Part]`*
RESERVECHECK

RESERVECHECK

WARN IF: *`QAccomdat.Tenure IN [Outright .. Part]`*
RESERVECHECK

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: (((QAccomdat.Tenure IN [Mortgage .. Part]) OR (QOwner1.OthMort3 =
     Yes)) AND (QOwner1.QMortgage.M[1].MorInPay = EMPTY OR
     (QOwner1.QMortgage.M[1].MorInPay > 0))) AND
     (QOwner1.QMortgage.M[2].MorInPay = EMPTY OR
     (QOwner1.QMortgage.M[2].MorInPay > 0))
```

```
AskStruc := 1
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: (((QAccomdat.Tenure IN [Outright .. Part]) OR QAccomdat.Tenure =
     NONRESPONSE) OR (QRenting.Landlord IN [Assocn .. OthIndiv])) OR
     QRenting.Landlord = NONRESPONSE
     AND: AskStruc = 1
```

```
AskStruc := 3
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: (((QAccomdat.Tenure IN [Outright .. Part]) OR QAccomdat.Tenure =
     NONRESPONSE) OR (QRenting.Landlord IN [Assocn .. OthIndiv])) OR
     QRenting.Landlord = NONRESPONSE
     AND: NOT (AskStruc = 1)
```

```
AskStruc := 2
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: (QAccomdat.Tenure = Part) AND (QAccomdat.SOBuy = Paid)
```

```
AskStruc := 2
```

FRS0405A.QInsur

Questions about structure insurance.

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
```

StrMort

QInsur

Did your last payment on the mortgage/ loan include an amount for any insurance on the structure of this accommodation, its furniture or contents, or any personal possessions?

(1) Yes
(2) No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
```

StrCov

QInsur

Was that for...READ OUT (RUNNING PROMPT)...

(1) ...structure ONLY
(2) ...furniture and contents or personal possessions, only
(3) ...structure AND furniture and contents, or personal possessions?

FRS0405A.QInsur.QStructure[]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: (PStrCov = Struct) OR (PCovOths = Struct)
```

combined := ' structure'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: PStrCov = Furn
```

combined := ' contents'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: (PStrCov = Combine) OR (PCovOths = Combine)
```

combined := ' combined'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: PSeq = 1
```

included := ' included in your last mortgage payment'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: NOT (PSeq = 1)
```

last := ' last'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
```

StrAmt

QInsur

How much was the^last premium^included for this^combined policy?

0.01..99997.00

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

StrPx

QInsur

^Pd97Txt

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

StrPd

QInsur

How long did this cover?

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*
AND: *StrPd = Note*

StrPx

QInsur

^Pd97Txt

OPEN

FRS0405A.QInsur.QStructure[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QInsur.QStructure[] (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

StrWkly := LWeekly

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (StrWkly < 50) AND INVOLVING(StrPd,StrAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     StrPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QInsur (continued)

Questions about structure insurance.

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: (StrMort = EMPTY OR (StrMort = No)) OR (StrCov = Furn)
```

StrOths

QInsur

Do you pay an insurance premium on the structure of this accommodation?

(1) Yes
(2) No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
```

CovOths

QInsur

Does the premium cover...READ OUT (RUNNING PROMPT)...

(1) ...structure ONLY,
(2) ...or structure combined with furniture, contents or personal possessions?

FRS0405A.QInsur.QStructure[]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: (PStrCov = Struct) OR (PCovOths = Struct)
```

combined := ' structure'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: PStrCov = Furn
```

combined := ' contents'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: (PStrCov = Combine) OR (PCovOths = Combine)
```

combined := ' combined'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: PSeq = 1
```

included := ' included in your last mortgage payment'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: NOT (PSeq = 1)
```

last := ' last'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
```

StrAmt

QInsur

How much was the^last premium^included for this^combined policy?

0.01..99997.00

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

StrPx

QInsur

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

StrPd

QInsur

How long did this cover?

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: StrPd = Note
```

StrPx

QInsur

^Pd97Txt

OPEN

FRS0405A.QInsur.QStructure[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0405A.QInsur.QStructure[] (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

StrWkly := LWeekly

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (StrWkly < 50) AND INVOLVING(StrPd,StrAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     StrPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QInsur (continued)

Questions about structure insurance.

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: QInsur.QStructure[1].StrWkly = RESPONSE
     AND: QOwner1.QMortgage.M[1].MorIWkly = RESPONSE
     (QInsur.QStructure[1].StrWkly < QOwner1.QMortgage.M[1].MorIWkly) AND
     INVOLVING(QOwner1.QMortgage.M[1].MorInPay,QInsur.QStructure[1].StrAmt,
     QInsur.QStructure[1].StrPd)
```

The amount you recorded for the premium on the insurance on the structure is greater than the amount recorded for the last mortgage payment.
Please check whether this is correct.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: QInsur.QStructure[1].StrWkly = RESPONSE
     AND: QOwner1.QMortgage.M[1].IntPWkly = RESPONSE
     (QInsur.QStructure[1].StrWkly < QOwner1.QMortgage.M[1].IntPWkly) AND
     INVOLVING(QOwner1.QMortgage.M[1].IntPrPay,QInsur.QStructure[1].StrAmt,
     QInsur.QStructure[1].StrPd)
```

The amount you recorded for the premium on the insurance on the structure is greater than the amount recorded for the last mortgage payment.
Please check whether this is correct.

FRS0405A.QCounTax

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandAMax := 1020

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandBMax := 1190

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandCMax := 1360

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandDMax := 1525

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandEMax := 1865

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandFMax := 2205

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandGMax := 2545

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandHMax := 3050

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

BandAMin := 400

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

**CTIntro := 'NOW THERE ARE SOME QUESTIONS ABOUT COUNCIL TAX
'**

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

CTConDoc

QCounTax

^CTIntro
For your Council Tax, do you have a bill, or a payment book that you could consult?

ACCEPT A STATEMENT/BILL FROM THE YEAR 2003-2004 IF NO PAYMENT FOR 2004-2005 YET MADE.

(1) Yes - consulted now

(2) No - no document (or will not consult)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

CTBand

QCounTax

Could you please tell me which Council Tax band this accommodation is in?

THIS MUST BE THE BAND GIVEN BY THE COUNCIL - DO NOT ACCEPT RESPONDENT'S OWN ESTIMATE OF VALUE OF PROPERTY.

IF THIS HOUSEHOLD'S ACCOMMODATION IS NOT VALUED SEPARATELY (eg. because it's a rented part of larger premises), THEN USE CODE 9.

INTERVIEWER: If respondents initial band allocation was later changed because they are disabled enter original band here.

(1) Band A
(2) Band B
(3) Band C
(4) Band D
(5) Band E
(6) Band F
(7) Band G
(8) Band H
(9) Household accommodation not valued separately

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

CTXAmt := 'Missing'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

CTSXAmt := 'Missing'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

CTSWAmt := 'Missing'

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*

CTSSAmt := 'Missing'

Warn if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *Edit = Yes*
    And: *CTConDoc <> EMPTY*
    CTBand = RESPONSE

EDITOR: THE COUNCIL TAX BAND IS MISSING. THE LOCAL AUTHORITY WILL NEED TO BE TELEPHONED. CONSULT THE FACT SHEET & TELEPHONE THEM.

Record if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*

CTValid

QCounTax

STRING[2]

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

Letters[1] := 'A'

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

Letters[2] := 'B'

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

Letters[3] := 'C'

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

Letters[4] := 'D'

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

Letters[5] := 'E'

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    And: *NatCen <> NI*
    And: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

Letters[6] := 'F'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

Letters[7] := 'G'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

Letters[8] := 'H'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTBand IN [BandA .. BandH]
```

Letter := Letters[ORD(CTBand)]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTBand = NotApp
```

Letter := 'Not valued separately'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTBand = DONTKNOW
```

Letter := 'Don't know'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTBand = REFUSAL
```

Letter := 'Missing'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CTLVBand

QCounTax

Was your Council Tax bill reduced to a lower band because there is a disabled person in the household?

INTERVIEWER: HOUSEHOLDS MUST MAKE A SPECIAL APPLICATION IN ORDER TO OBTAIN THIS REDUCTION.

(1) Yes
(2) No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (CTLVBand = Yes) AND (CTBand IN [BandA .. BandH])
```

CTLVChk

QCounTax

You said you were in Band ^Letter; is that the Band after this lower valuation, or before?

(1) After lower valuation
(2) Before

```
CHECK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (CTLVBand = Yes) AND (CTBand IN [BandA .. BandH])
     AND: CTBand = BandH
     CTLVChk <> Aftr
```

Band H is the HIGHEST band, so it cannot be the band AFTER the lower valuation. Please change one or the other.

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: Scotland = Yes
```

ScotFill := (' including Domestic water & sewerage' + ' charges')

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: NOT (Scotland = Yes)
```

ScotFill := ''

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CTAmt

QCounTax

If the respondent has not paid any tax for any reason then enter 0 and the later questions will probe the reasons.

0.00..9999.97

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: Scotland = Yes
     CTAmt <> 0
```

No Council Tax paid, but you should record here the last payment of domestic water charge and domestic sewerage charge, which are not paid for by C. Tax benefit. If they have not been paid, suppress warning and continue.

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *CTAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *(CTAmt > 0) OR CTAmt = NONRESPONSE*

CTInstal

QCounTax

(Can I just check,) Was that the full payment for the year, or was it an instalment?

INTERVIEWER: 'YEAR' = APRIL TO MARCH (12 MONTHS).

(1) Full annual payment
(2) An instalment

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *(CTAmt > 0) OR CTAmt = NONRESPONSE*
AND: *CTInstal = Instal*

CTTime

QCounTax

How many instalments are there, over the whole year?

'WHOLE YEAR' = APRIL TO MARCH (12 MONTHS).
IF PAYMENT GIVEN IS FROM LAST YEAR, ENTER NUMBER OF INSTALMENTS MADE LAST YEAR.

2..52

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *(CTAmt > 0) OR CTAmt = NONRESPONSE*
AND: *CTInstal = Instal*
AND: *(CTConDoc = Yes) AND (CTTime = RESPONSE)*

CTAnnual

QCounTax

REFER TO DOCUMENT BEING CONSULTED:

On the statement/bill, what is the total amount payable for the year,^ScotFill after deducting any discounts or benefit?

'YEAR' = APRIL TO MARCH (12 MONTHS)

0.00..9999.97

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: (CTAmt = RESPONSE) AND (CTInstal = RESPONSE)
      AND: CTInstal = Full
```

```
CTAmtYr := CTAmt
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: (CTAmt = RESPONSE) AND (CTInstal = RESPONSE)
      AND: (CTInstal = Instal) AND (CTTime = RESPONSE)
```

```
CTAmtYr := (CTAmt * CTTime)
```

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CWat1Rs

QCounTax

In Scotland, Domestic Water Charge should be included in the total Council Tax bill for the year - if not, please explain in a note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CWat1Ex

QCounTax

^SuppTxt

OPEN

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CSew1Rs

QCounTax

In Scotland, Domestic Sewerage Charge should be included in the total Council Tax bill for the year - if not, please explain in a note.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

CSew1Ex

QCounTax

^SuppTxt

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*

CWatAmt1

QCounTax

How much is the annual Domestic Water Charge, as shown on the bill?

INTERVIEWER: ENTER THE FULL CHARGE, BEFORE ANY STATUS DISCOUNT.

0.00..999.97

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*
CWatAmt1 > 0

CWatAmt1 > 0

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*
AND: *(CWat1Rs = Suppressed) OR CWat1Ex <> EMPTY*

CWat1Ex

QCounTax

^SuppTxt

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND
     ((CTInstal = Full) OR (CTAnnual > 0))
     AND: CWatAmt1 = RESPONSE
```

CSewAmt1

QCounTax

How much is the annual Domestic Sewerage Charge, as shown on the bill?

INTERVIEWER: ENTER THE FULL CHARGE, BEFORE ANY STATUS DISCOUNT OR TRANSITIONAL RELIEF.

0.00..999.97

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND
     ((CTInstal = Full) OR (CTAnnual > 0))
     AND: CWatAmt1 = RESPONSE
     AND: CSewAmt1 = RESPONSE
     CSewAmt1 > 0
```

CSewAmt1 > 0

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND
     ((CTInstal = Full) OR (CTAnnual > 0))
     AND: CWatAmt1 = RESPONSE
     AND: CSewAmt1 = RESPONSE
     AND: (CSew1Rs = Suppressed) OR CSew1Ex <> EMPTY
```

CSew1Ex

QCounTax

^SuppTxt

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: ((Edit = Yes) AND (Scotland = Yes)) AND (CTConDoc = Yes)
```

CWatAmt1

QCounTax

How much is the annual Domestic Water Charge, as shown on the bill?

INTERVIEWER: ENTER THE FULL CHARGE, BEFORE ANY STATUS DISCOUNT.

0.00..999.97

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: ((Edit = Yes) AND (Scotland = Yes)) AND (CTConDoc = Yes)
```

CSewAmt1

QCounTax

How much is the annual Domestic Sewerage Charge, as shown on the bill?

INTERVIEWER: ENTER THE FULL CHARGE, BEFORE ANY STATUS DISCOUNT OR TRANSITIONAL RELIEF.

0.00..999.97

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CTRebPx

QCounTax

^Pd97Txt

OPEN

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CTRebRs

QCounTax

That seems rather high. Please check the amount and frequency of payment. If correct, suppress warning and explain circumstances in a NOTE.

(1) Passed
(2) Hard
(3) Soft
(4) Suppressed

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CTRebEx

QCounTax

^SuppTxt

OPEN

***ASK IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
*　　**AND:** NatCen <> NI*
*　　**AND:** (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*

CTReb

QCounTax

Are you allowed Council Tax Benefit or rebate, to help pay your Council Tax?

(1) Yes
(2) No

***ASK IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
*　　**AND:** NatCen <> NI*
*　　**AND:** (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
*　　**AND:** CTReb = No*
*　　**AND:** (CTAmt = 0) OR CTAmt = NONRESPONSE*

WhyNoCT

QCounTax

INTERVIEWER: NO COUNCIL TAX IS PAID, BUT NO BENEFIT RECEIVED.
ASK OR CODE: WHAT WAS THE REASON FOR PAYING NO COUNCIL TAX?

(1) Bill not yet received and household not previously liable for C.Tax
(2) Bill not yet paid and household not previously liable for C.Tax
(3) Deliberate non-payment, in dispute, appeal, etc.
(4) Household only recently moved into accommodation
(5) Household has a 'formal exemption' from the Tax
(all students; MoD property; severely mentally impaired.)
(6) Other reason (DESCRIBE IN A NOTE)

***ASK IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
*　　**AND:** NatCen <> NI*
*　　**AND:** (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
*　　**AND:** CTReb = No*

CTBWait

QCounTax

Are you awaiting the outcome of a claim for Council Tax benefit or rebate?

(1) Yes
(2) No

***ASK IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
*　　**AND:** NatCen <> NI*
*　　**AND:** (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
*　　**AND:** CTReb = Yes*

CTRebAmt

QCounTax

How much was allowed?

0.00..9999.97

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

CTRebPd

QCounTax

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd = Note
```

CTRebPx

QCounTax

^Pd97Txt

OPEN

FRS0405A.QCounTax.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[1] := 1
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[2] := 2
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[3] := 3
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[4] := 4
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[5] := 4.333
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[7] := 8.67
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0

PdConW[8] := 6.5
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0

PdConW[9] := 5.78
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0

PdConW[10] := 5.2
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0

PdConW[13] := 13
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0

PdConW[26] := 26
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0

PdConW[52] := 52
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0
      AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])

PWeekly := (PAmount / PdConW[ORD(PPeriod)])
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NatCen <> NI
      AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
      AND: CTReb = Yes
      AND: CTRebAmt > 0
      AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])

PWeekly := 0
```

FRS0405A.QCounTax (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

CTRWkly := LWeekly

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

CTRebYr := (CTRWkly * 52)

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (CTRebYr > 0) AND (CTBand = RESPONSE)
     ((((((((CTBand = BandA) AND (CTRebYr <= BandAMax)) OR ((CTBand =
     BandB) AND (CTRebYr <= BandBMax))) OR ((CTBand = BandC) AND (CTRebYr
     <= BandCMax))) OR ((CTBand = BandD) AND (CTRebYr <= BandDMax))) OR
     ((CTBand = BandE) AND (CTRebYr <= BandEMax))) OR ((CT
```

((((((((CTBand = BandA) AND (CTRebYr <= BandAMax)) OR ((CTBand = BandB) AND (CTRebYr <= BandBMax))) OR ((CTBand = BandC) AND (CTRebYr <= BandCMax))) OR ((CTBand = BandD) AND (CTRebYr <= BandDMax))) OR ((CTBand = BandE) AND (CTRebYr <= BandEMax))) OR ((CTBand = BandF) AND (CTRebYr <= BandFMax))) OR ((CTBand = BandG) AND (CTRebYr <= BandGMax))) OR ((CTBand = BandH) AND (CTRebYr <= BandHMax))) AND INVOLVING (CTBand, CTRebPd, CTRebAmt)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (CTRebYr > 0) AND (CTBand = RESPONSE)
     AND: (CTRebRs = Suppressed) OR CTRebEx <> EMPTY
```

CTRebEx

QCounTax

^SuppTxt

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (((NewBU >= 2) AND (CTConDoc = Yes)) AND (CTReb = Yes)) AND
     (NotHRPBU = 1)
```

WhoseCTB

QCounTax

According to the statement, who is the Council Tax Benefit for?

CODE ALL THAT APPLY.

SET [7] OF
(1) ^BUAdName[1]
(2) ^BUAdName[2]
(3) ^BUAdName[3]
(4) ^BUAdName[4]
(5) ^BUAdName[5]
(6) ^BUAdName[6]
(7) ^BUAdName[7]
(8) Someone else (SPECIFY IN A NOTE)
(9) Not on statement

```
CHECK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (((NewBU >= 2) AND (CTConDoc = Yes)) AND (CTReb = Yes)) AND
     (NotHRPBU = 1)
     AND: In loop FOR Index := 1 TO 7
     AND: Index IN WhoseCTB
     BUAdName[[Index] <> ''
```

Code ^Index is not valid for this question.

```
CHECK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: (((NewBU >= 2) AND (CTConDoc = Yes)) AND (CTReb = Yes)) AND
     (NotHRPBU = 1)
     AND: NS IN WhoseCTB
     WhoseCTB.CARDINAL = 1
```

'Not known/not on statement' is an exclusive code!

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
```

are := 'In addition to your rebate/ benefit, are'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: NOT (CTReb = Yes)
```

are := 'Are'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: NatCen <> NI
```

SHOWCARD := 'SHOW CARD O'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: NOT (NatCen <> NI)
```

SHOWCARD := ''

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
```

CTDisc

QCounTax

The council tax assumes two adults per household. If only one adult lives there, a 25% status discount will usually apply. If a student/student nurse/apprentice etc. lives with one other adult then a discount of 25% will apply. However no discount will apply if that person lives with two or more other adults.

(1) Yes
(2) No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE
     AND: CTDisc = Yes
```

CT25D50D

QCounTax

Some households get a discount on their Council Tax because of the TYPE of person living there. Usually this is a 25% discount. The rules are very detailed, but in general:

25% for Single adult households

25% for one adult, plus: a student/student nurse/person under 18/apprentice/YT Trainee/care worker/severely mentally impaired

50% when a household is made up of a mixture of those who are exempt (eg. student/student nurse/person under 18/apprentice/YT Trainee/care worker/severely mentally impaired)

Note: Households wholly occupied by students or under 18 year olds are entirely EXEMPT from Council Tax.

(1) 25%
(2) 50%

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *CTDisc = Yes*
CT25D50D <> D50

Are you sure? Discount is usually 25%. The 50% discount applies only if ALL household members belong to the groups shown on ^SHOWCARD. Please check with respondent. If discount IS DEFINITELY 50%, suppress warning and continue.

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandH]) OR CTBand = NONRESPONSE*
AND: *AllAd = 1*
(CTDisc = Yes) AND (CT25D50D = D25)

Are you sure? Households with only one adult would normally have a status discount (25% reduction of the bill).

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTAmtYr > 0) AND (CTBand = RESPONSE)*
((((((((CTBand = BandA) AND (CTAmtYr <= BandAMax)) OR ((CTBand = BandB) AND (CTAmtYr <= BandBMax))) OR ((CTBand = BandC) AND (CTAmtYr <= BandCMax))) OR ((CTBand = BandD) AND (CTAmtYr <= BandDMax))) OR ((CTBand = BandE) AND (CTAmtYr <= BandEMax))) OR ((CT

That's ^P^CTAmtYr a year which seems rather high for a property in this Band. Please check the amount and frequency of payment. If correct, suppress warning and explain circumstances in a Note.

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *((CTAmt > 0) AND (CTDisc <> Yes)) AND (CTReb <> Yes)*
((CTInstal = Instal) AND ((CTAmt * CTTime) >= BandAMin)) OR ((CTInstal = Full) AND (CTAmt > BandAMin))

The annual Council Tax comes to less than the cheapest Council Tax. No discount or rebate is received, so please check for a typing error. If correct, please give explanation in a Note.

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTAnnual =
     RESPONSE)
```

CTReal := (CTAmt * CTTime)

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTAnnual =
     RESPONSE)
     CTAnnual < (1.2 * CTReal)
```

^P^CTAnnual is more than would be expected, given the instalments mentioned earlier. Please check, from the document consulted, that it's for the same year as the instalments.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTAnnual =
     RESPONSE)
     CTAnnual > (0.8 * CTReal)
```

^P^CTAnnual is less than would be expected, given the instalments mentioned earlier. Please check, from the document consulted, that it's for the same year as the instalments.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     (CTBand <> NotApp) AND (CTBand <> DONTKNOW)
```

INTERVIEWER: if necessary check which is correct; this accom:

- is NOT valued separately for C.Tax (code 9),
- or it IS valued for C.Tax, but respondent DOESN'T KNOW the Tax Band
(enter Don't know).

If correct, suppress check & continue.

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTInstal =
     Instal)) AND (CTAnnual = RESPONSE)) AND (CWatAmt1 = RESPONSE)
```

CTReal := (CTAmt * CTTime)

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTInstal =
     Instal)) AND (CTAnnual = RESPONSE)) AND (CWatAmt1 = RESPONSE)
     CWatAmt1 <= CTReal
```

The Domestic Water charge is more than the total amount of Council Tax paid for the year. Make sure that the last payment of Domestic Water charge (and domestic sewerage charge) was included at CTAmt.

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CWatAmt1 = RESPONSE) AND (CTAmt = RESPONSE)) AND (CTInstal =
     Full)
     CWatAmt1 <= CTAmt
```

The Domestic Water Charge is more than the total amount of Council Tax paid for the year. Make sure that the last payment of Domestic Water charge (and domestic sewerage charge) was included at CTAmt.

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    AND: *CTReb <> EMPTY AND (CTRebAmt = RESPONSE)*
    CTRebAmt <> 0

Zero amount of Council Tax benefit: this contradicts previous answer (at 'CTReb') that benefit WAS received. Please resolve if possible.

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
QCounTax.WhyNoCT <> Other

If Tax is 'paid by landlord' or 'included in rent', check whether the accommodation is valued separately and if not, recode CTBand;
if a deduction is made from pay, enter it as a payment at CTAmt.

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *QCounTax.CTReb <> EMPTY AND ((QRenting.HBenefit = Yes) OR (GOV IN QOwner1.QMortgage.M[1].QOutsPay))*
QCounTax.CTReb = Yes

Earlier, the respondent said they get Housing Benefit or help from ^GOV1 with mortgage. They should usually also get Council Tax Benefit. Please check: is their C.Tax bill 'reduced' - does the Council take anything off it? (apart from Discount). If so, change answer to 'Yes'.

FRS0405A.QNIRates

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 1

NIRate := 2.8728
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 2

NIRate := 2.9465
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 3

NIRate := 3.0742
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 4

NIRate := 2.7313
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 5

NIRate := 2.8478
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 6

NIRate := 2.9397
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 7

NIRate := 3.0763
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 8

NIRate := 2.9691
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 9

NIRate := 2.6758
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 10
```

NIRate := 2.7528

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 11
```

NIRate := 2.8189

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 12
```

NIRate := 2.9642

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 13
```

NIRate := 3.0007

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 14
```

NIRate := 2.7366

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 15
```

NIRate := 2.6023

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 16
```

NIRate := 3.0383

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 17
```

NIRate := 2.871

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 18
```

NIRate := 2.8338

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 19
```

NIRate := 3.1045

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 20
```

NIRate := 2.64

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 21
```

NIRate := 3.0932

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 22
```

NIRate := 3.0493

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 23
```

NIRate := 2.9271

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 24
```

NIRate := 2.8797

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 25
```

NIRate := 3.0022

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 26
```

NIRate := 2.8805

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[1] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[2] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[3] := 17.33

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[4] := 13

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[5] := 12

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[7] := 6

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[8] := 8

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[9] := 9

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[10] := 10

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[13] := 4

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[26] := 2

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[52] := 1

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[90] := 1

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

PDCode[95] := 1

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*

AND: *NOT (NatCen <> NI)*

RTIntro := 'Now there are some questions about Rates
'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

BillRate

^RTIntro
Do you get a bill for rates on this accommodation?

(1) Yes
(2) No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: BillRate = No
```

NoRate

Why do you not get a rates bill?

(1) Rented accommodation with rates included in rent
(2) Rent/rates free
(3) Receive rebate
(4) Other reason (specify)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: BillRate = No
     AND: NoRate = Other
```

OthReas

Please specify this other reason

STRING[100]

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
```

PayRate

Do you, or someone in this household, pay the rates bill?

(1) Yes
(2) No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = No
```

NoPay

Why don't you pay your rates bill?

STRING[100]

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
```

RTConDoc

For your Rates, do you have a bill, or a payment book that you could consult?

ACCEPT A STATEMENT/BILL FROM THE YEAR 2001-2002 IF NO PAYMENT FOR 2002-2003 YET MADE.

(1) Yes - consulted now

(2) No - no document (or will not consult)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
     AND: RTConDoc = Yes
```

RTAnnual

REFER TO DOCUMENT BEING CONSULTED:

On the statement/bill, what is the total amount payable, after deducting any discounts or benefit?

'YEAR' = APRIL TO MARCH (12 MONTHS)

NOTE : NO RATES ARE PAYABLE IN FEBRUARY AND MARCH EACH YEAR

0.00..9999.97

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
     AND: RTConDoc = No
```

EstRTAnn

Can you tell me, what is the total amount of rates payable, after deducting any discounts or benefit?

INTERVIEWER: PROBE TO ENSURE AMOUNT GIVEN IS AS ACCURATE AS POSSIBLE.
IF RESPONDENT UNSURE OF AMOUNT OPEN A NOTE TO STATE VALUE IS AN ESTIMATE.

'YEAR' = APRIL TO MARCH (12 MONTHS)

NOTE : NO RATES ARE PAYABLE IN FEBRUARY AND MARCH EACH YEAR

0.00..9999.97

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
```

RTInstal

(Can I just check,) Was that the full payment for the year, or was it an instalment?

INTERVIEWER: 'YEAR' = APRIL TO MARCH (12 MONTHS).

(1) Full annual payment
(2) An instalment

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
     AND: RTInstal = Instal
```

RTTimePx

^Pd97Txt

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
     AND: RTInstal = Instal
```

RTTimePd

How often do you pay instalments?

'WHOLE YEAR' = APRIL TO MARCH (12 MONTHS).

NOTE : THERE ARE USUALLY 10 MONTHLY INSTALMENTS PER YEAR AS FEBRUARY AND MARCH ARE 'FREE' MONTHS.
IF PAYMENT GIVEN IS FROM LAST YEAR, ENTER NUMBER OF INSTALMENTS MADE LAST YEAR.

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *NOT (BillRate = No)*
AND: *PayRate = Yes*
AND: *RTInstal = Instal*
AND: *RTTimePd = Note*

RTTimePx

^Pd97Txt

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *NOT (BillRate = No)*
AND: *PayRate = Yes*

RTReb

Are you allowed a Rates Rebate?

(1) Yes
(2) No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*

RTDeduc

Was this deducted from your LAST rates payment?

(1) Yes
(2) No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*
AND: *RTDeduc = Yes*

RTRebAmt

How much was allowed?

0.01..9999.97

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*
AND: *RTDeduc = Yes*

RTRebPx

^Pd97Txt

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*
AND: *RTDeduc = Yes*

RTRebPd

How long did this cover?

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*
AND: *RTDeduc = Yes*
AND: *RTRebPd = Note*

RTRebPx

^Pd97Txt

OPEN

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *((RTAnnual = RESPONSE) AND (RTInstal <> Full)) AND (RTTimePd IN [OneWeek .. LessWeek])*

RTCheck := (RTAnnual * PDCode[ORD(RTTimePd)])

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *((EstRTAnn = RESPONSE) AND (RTInstal <> Full)) AND (RTTimePd IN [OneWeek .. LessWeek])*

RTCheck := (EstRTAnn * PDCode[ORD(RTTimePd)])

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(RTAnnual = RESPONSE) AND (RTInstal = Full)*

RTCheck := RTAnnual

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(EstRTAnn = RESPONSE) AND (RTInstal = Full)*

RTCheck := EstRTAnn

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: QDataBag.NINRV > 0
     AND: (RTAnnual = RESPONSE) OR (EstRTAnn = RESPONSE)
     (RTCheck < (QDataBag.NINRV * NIRate)) AND
     INVOLVING(RTInstal,RTAnnual,EstRTAnn)
```

That's ^P^RTCheck per year for Rates which seems high for a property in this area. Are you sure the Amount of Rates paid and the Period are correct?

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
***AND:** QNIRates.RTReb <> EMPTY AND ((QRenting.HBenefit = Yes) OR (GOV IN QOwner1.QMortgage.M[1].QOutsPay))*
QNIRates.RTReb = Yes

Earlier, the respondent said they get Housing Benefit or help from ^GOV1 with mortgage. They should usually also get a Rates Rebate. Please check: is their Rates bill 'reduced' - does the Rate Collection Agency take anything off it? If so, change answer to 'Yes'.

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
***AND:** QNIRates.BillRate <> EMPTY AND (QNIRates.NoRate = RateInc)*
((QAccomdat.Tenure = Part) OR (QAccomdat.Tenure = Rents)) AND INVOLVING(QAccomdat.Tenure)

Earlier, the respondent said they did NOT rent/part rent this accommodation - the rates could NOT be included in the rent!!
Please check that Tenure is correct.

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

***WARN IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
  AND: *NOT (NatCen <> NI)*
  RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
  AND: *NOT (NatCen <> NI)*
  RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
  RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
  RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
  RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
  RESERVECHECK

RESERVECHECK

COMPUTE IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
  AND: *(QRenting.WSInc = EMPTY OR QRenting.WSInc = NONRESPONSE) OR (QRenting.WSInc IN [Sewer, Neith])*

AskWater := Yes

COMPUTE IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
  AND: *(QRenting.WSInc = EMPTY OR QRenting.WSInc = NONRESPONSE) OR (QRenting.WSInc IN [Water, Neith])*

AskSewer := Yes

FRS0405A.QWaterSew

Questions about sewerage and water rates

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: PAskWater = Yes
```

WaterMet

QWaterSew

Are your water charges metered or not?

(1) Yes
(2) No

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: PAskWater = Yes
```

WaterPay

QWaterSew

Do you pay water rates or charges?

(1) Yes
(2) No

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: PAskSewer = Yes
```

SewerPay

QWaterSew

Do you pay sewerage rates or charges?

(1) Yes
(2) No

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (WaterPay = Yes) AND (SewerPay = Yes)
```

SewSep

QWaterSew

Do you pay separate or combined water and sewerage rates or charges?

(1) Separate
(2) Combined

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
```

WatTime

QWaterSew

How many times a year do you pay water rates or charges?

ENTER TIMES A YEAR.

1..52

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
```

WatAmt

QWaterSew

How much did you actually pay last time?

0.01..9997.00

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
```

WatAnul

QWaterSew

How much is your annual bill?

0.01..9997.00

```
COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
     AND: WatAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
     AND: (WatAmt = RESPONSE) AND (WatTime = RESPONSE)
```

WatWkly := ((WatAmt * WatTime) / 52)

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
     AND: Edit = No
     (WatWkly <= 12) AND INVOLVING(WatTime,WatAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

WARN IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
AND: *(SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))*
AND: *Edit = No*
AND: *((WatTime = RESPONSE) AND (WatAnul = RESPONSE)) AND (WatAmt = RESPONSE)*
(ABS((WatTime * WatAmt) - WatAnul) <= 25) AND INVOLVING(WatTime,WatAnul,WatAmt)

INTERVIEWER: The Annual payment for water rates/charges (WatAnul) is very different from the total for individual payments (WatTime x WatAmt).
Please check these figures.

ASK IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
AND: *(SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*

SewTime

QWaterSew

How many times a year do you pay sewerage rates or charges?

ENTER TIMES A YEAR.

1..52

ASK IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
AND: *(SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*

SewAmt

QWaterSew

How much did you actually pay last time?

0.01..9997.00

ASK IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
AND: *(SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*

SewAnul

QWaterSew

How much is your annual bill?

0.01..9997.00

COMPUTE IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
AND: *(SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*
AND: *SewAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))
     AND: (SewAmt = RESPONSE) AND (SewTime = RESPONSE)
```

SewWkly := ((SewAmt * SewTime) / 52)

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))
     AND: Edit = No
     (SewWkly <= 8) AND INVOLVING(SewTime,SewAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))
     AND: Edit = No
     AND: ((SewTime = RESPONSE) AND (SewAnul = RESPONSE)) AND (SewAmt =
     RESPONSE)
     (ABS((SewTime * SewAmt) - SewAnul) <= 25) AND
     INVOLVING(SewTime,SewAnul,SewAmt)
```

INTERVIEWER: The Annual payment for sewerage rates/charges (SewAnul) is very different from the total for individual payments (SewTime x SewAmt).
Please check these figures.

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
```

WSewTime

QWaterSew

How many times a year do you pay?

ENTER TIMES A YEAR.

1..52

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
```

WSewAmt

QWaterSew

Metered Water - Charges made via a water meter should be treated as water rate payments and the last amount actually paid entered.

0.01..9997.00

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
```

WSewAnul

QWaterSew

How much is your annual bill?

0.01..9997.00

```
COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: WSewAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: (WSewAmt = RESPONSE) AND (WSewTime = RESPONSE)
```

WSewWkly := ((WSewAmt * WSewTime) / 52)

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: Edit = No
     (WSewWkly < 20) AND INVOLVING(WSewTime,WSewAmt)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: Edit = No
     AND: ((WSewTime = RESPONSE) AND (WSewAnul = RESPONSE)) AND (WSewAmt =
     RESPONSE)
     (ABS((WSewTime * WSewAmt) - WSewAnul) <= 25) AND
     INVOLVING(WSewTime,WSewAnul,WSewAmt)
```

INTERVIEWER: The Annual payment for water/sewerage rates/charges (WSewAnul) is very different from the total for individual payments (WSewTime x WSewAmt).
Please check these figures.

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: WaterMet = Yes
```

WatRb

QWaterSew

The vast majority of people have to pay the full water and sewage charges but there are also a few metered who are eligible for assistance under the Vulnerable Groups Scheme (Water Industry Act 1999). Under this scheme people who qualify receive a bill capped at the average charge for their region and do not have to pay the measured charge reflecting their genuine water consumption.

(1) Yes
(2) No

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QAccomCharge

Questions on charges with accommodation.

Ask if: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*

Charge

QAccomCharge

Use the combined charge category when it is not possible for respondents to split out separate amounts for ground rent, service charge and maintenance charge. If you record a combined amount, do not record the separate amounts as well.

SET [9] OF
(1) Ground Rent
(2) Feu duty
(3) Chief Rent
(4) Service charge
(5) Compulsory or regular maintenance charges
(6) Site rent (caravans)
(7) Factoring (Payments to a land steward)
(8) Any other regular payments
(9) Combined charges (eg. ground rent, service charge, maintenance charge, factoring etc.)
(10) None of these

Warn if: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
And: *(((Ground IN Charge) AND (Service IN Charge)) AND (Maint IN Charge)) AND (Combined IN Charge)*
ERROR

Only code combined charges instead of the separate ground rent, service charge, maintenance charge, factoring etc.

Ask if: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
And: *Other IN Charge*

ChargeO

QAccomCharge

Please specify the other type of payments.

STRING[50]

Ask if: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
And: *(Service IN Charge) OR (Combined IN Charge)*

ChIns

QAccomCharge

Does this service charge include insurance?

(1) Yes
(2) No

FRS0405A.QAccomCharge.QChargeAmtPd[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[1] := 'Ground Rent'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[2] := 'Feu duty'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[3] := 'Chief Rent'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[4] := 'Service charge'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[5] := 'Maintenance charges'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[6] := 'Site rent'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[7] := 'Factoring'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[8] := PChargeO

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[9] := ('Combined charges (eg. ground rent, service charge, ' + 'maintenance charge, factoring etc.)')

ASK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *In loop FOR Idx := 1 TO 9*
AND: *Idx IN Charge*

ChrgAmt

QAccomCharge

I would now like to ask about the charges you pay for ^LCharges[PSeq].
How much did you pay last time?

0.01..9997.00

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *In loop FOR Idx := 1 TO 9*
AND: *Idx IN Charge*
AND: *ChrgAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *In loop FOR Idx := 1 TO 9*
AND: *Idx IN Charge*
AND: *ChrgAmt > 0*

ChrgPx

QAccomCharge

^Pd97Txt

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *In loop FOR Idx := 1 TO 9*
AND: *Idx IN Charge*
AND: *ChrgAmt > 0*

ChrgPd

QAccomCharge

How long did this cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: ChrgAmt > 0
     AND: ChrgPd = Note
```

ChrgPx

QAccomCharge

^Pd97Txt

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: Edit = Yes
     ChrgPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QAccomCharge (continued)

Questions on charges with accommodation.

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: FeuDuty IN Charge
     Scotland = Yes
```

Feu duty is only valid for Scottish households.

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: None IN Charge
     Charge.CARDINAL = 1
```

'None of these' is an exclusive code for this question.

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
RESERVECHECK

RESERVECHECK

FRS0405A.QLodger

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *In loop FOR count := 1 TO HHSize*
AND: *((PRec[count].Depend IN [Adult .. DepAd]) AND (PRel.PR[count].R IN [Child .. NonRel])) AND (ECount < 5)*

ECount := (ECount + 1)

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *In loop FOR count := 1 TO HHSize*
AND: *((PRec[count].Depend IN [Adult .. DepAd]) AND (PRel.PR[count].R IN [Child .. NonRel])) AND (ECount < 5)*

ELodger[ECount] := count

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*

BordLodg[count].BenUnit := DMBU[[ELodger[count]]

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*

BordLodg[count].PersId := ELodger[count]

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRel.PR[ELodger[count]].R IN [FChild, FParent, FSib, GChild .. NonRel]*

Relation := Distant

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRec[ELodger[count]].Depend = DepAd*

Relation := Skip

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRel.PR[ELodger[count]].R IN [Child .. StChild, ILChild ..
     StParent, ILParent .. StSib, ILSib]
```

Relation := Close

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRec[ELodger[count]].Sex = Male
```

HeShe := 'he'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: NOT (PRec[ELodger[count]].Sex = Male)
```

HeShe := 'she'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
```

LName := DMName[[ELodger[count]]

FRS0405A.QLodger.BordLodg[]

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*

BenUnit

QLodger

Benefit Unit of respondent.

0..7

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*

PersId

QLodger

Person number of respondent.

0..14

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*

ConvBL

QLodger

(Can I just check), is ^LName ...READ OUT (RUNNING PROMPT)...

(1) ...a BOARDER: that is, someone who pays you a RENT for board AND lodging
(2) ...a LODGER: that is, someone who pays you a RENT for lodging, but not food
(3) ...or neither of these?

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL = Board
```

pay := ' pay for board and lodging'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL = Lodg
```

pay := ' pay'

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
```

CvPay

QLodger

How much rent did ^LName^pay last time it was due, after deducting any Housing Benefit?

0.00..997.00

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay = NONRESPONSE
```

HMissVar := (HMissVar + 1)

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

CvPx

QLodger

^Pd97Txt

OPEN

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

CvPd

QLodger

How long does that cover?

(1) One week
(2) Two weeks
(3) Three weeks
(4) Four weeks
(5) Calendar month
(7) Two Calendar months
(8) Eight times a year
(9) Nine times a year
(10) Ten times a year
(13) Three months/13 weeks
(26) Six months/26 weeks
(52) One Year/12 months/52 weeks
(90) Less than one week
(95) One off/lump sum
(97) None of these (EXPLAIN IN A NOTE)

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
     AND: CvPd = Note
```

CvPx

QLodger

^Pd97Txt

OPEN

FRS0405A.QLodger.BordLodg[].Weekly()

Procedure Call

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[1] := 1

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[2] := 2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[3] := 3

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[4] := 4

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[5] := 4.333

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[7] := 8.67

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[8] := 6.5

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[9] := 5.78

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[10] := 5.2

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[13] := 13

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[26] := 26

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

PdConW[52] := 52

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*
AND: *(PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*
AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS0405A.QLodger.BordLodg[] (continued)

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
     AND: CvPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

CvWkly := LWeekly

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: (ConvBL = Lodg) AND (CvPay > 0)
```

CvHt

QLodger

Is HEATING included in that, or is it paid for separately?

(1) Included
(2) Paid for separately

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     CvPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

FRS0405A.QLodger (continued)

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: Edit = No
     AND: BordLodg[count].CvWkly = RESPONSE
     (BordLodg[count].CvWkly < 119) AND
     INVOLVING(BordLodg[count].CvPd,BordLodg[count].CvPay)
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *In loop FOR Loop1 := 1 TO 5*
AND: *QLodger.BordLodg[Loop1].CvPay > 0*

BUHBElig[QLodger.BordLodg[Loop1].BenUnit] := Yes

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
RESERVECHECK

RESERVECHECK

FRS0405A.QSharer

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO HHSize*
AND: *(PRec[count].Depend = Adult) AND (ECount < 8)*

ECount := (ECount + 1)

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO HHSize*
AND: *(PRec[count].Depend = Adult) AND (ECount < 8)*

ESharer[ECount] := count

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*

Sharer[count].BenUnit := DMBU[[ESharer[count]]

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*

Sharer[count].PersId := ESharer[count]

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*

LName := DMName[[ESharer[count]]

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *PRec[ESharer[count]].Sex = Male*

HeShe := 'he'

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *NOT (PRec[ESharer[count]].Sex = Male)*

HeShe := 'she'

FRS0405A.QSharer.Sharer[]

```
RECORD IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
```

BenUnit

QSharer

BU number of person

0..7

```
RECORD IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
```

PersId

QSharer

Person identifier.

0..14

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: PersId = 1
```

Preamb := ('Now I'd like to ask how much each of you' + ' pays towards certain things.')

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: NOT (PersId = 1)
```

Preamb := ''

Ask if: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
And: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
And: *In loop FOR count := 1 TO 8*
And: *ESharer[count] > 0*
And: *BenUnit > 1*

SRentAmt

QSharer

How much rent did ^LName pay last time it was due, after deducting any Housing Benefit?

0.00..997.00

Compute if: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
And: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
And: *In loop FOR count := 1 TO 8*
And: *ESharer[count] > 0*
And: *BenUnit > 1*
And: *SRentAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

Record if: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
And: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
And: *In loop FOR count := 1 TO 8*
And: *ESharer[count] > 0*
And: *BenUnit > 1*
And: *SRentAmt > 0*

SRentPx

QSharer

^Pd97Txt

OPEN

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*

SRentPd

QSharer

How long does that cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*
AND: *SRentPd = Note*

SRentPx

QSharer

^Pd97Txt

OPEN

FRS0405A.QSharer.Sharer[].Weekly()

Procedure Call

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[52] := 52

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*
AND: *(PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*
AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS0405A.QSharer.Sharer[] (continued)

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
     AND: SRentPd IN [OneWeek .. Year]
     AND: LWeekly >= 0
```

SRntWkly := LWeekly

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     SRentPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit = 1
```

How := 'Apart from any rent, how'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: NOT (BenUnit = 1)
```

How := 'How'

FRS0405A.QSharer (continued)

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: Edit = No
     AND: Sharer[count].SRntWkly = RESPONSE
     (Sharer[count].SRntWkly < 103) AND
     INVOLVING(Sharer[count].SRentPd,Sharer[count].SRentAmt)
```

Are you sure? Enter here only the RESPONDENT'S SHARE of the household rent.

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

FRS0405A.QProperty

Questions about other property

ASK IF: QAccomdat.SubLet = Yes

SubRent

QProperty

You mentioned earlier that you let, or sub-let, part of this accommodation to someone outside your household.

How much rent have you received from this in the last 12 months, ie. since ^DLYear : that's BEFORE deducting any income tax that might be due on it?

0.00..99999.97

COMPUTE IF: QAccomdat.SubLet = Yes
AND: SubRent = RESPONSE

SubWkly := (SubRent / 52)

WARN IF: QAccomdat.SubLet = Yes
AND: Edit = No
(SubWkly < 180) AND INVOLVING(SubRent)

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

COMPUTE IF: QAccomdat.SubLet = Yes
AND: SubRent = NONRESPONSE

HMissVar := (HMissVar + 1)

ASK IF: QAccomdat.SubLet = Yes

SubAllow

QProperty

And is that BEFORE or AFTER deducting allowable expenses?

(1) Before
(2) After

COMPUTE IF: QAccomdat.SubLet = Yes

Im := 'Apart from that, in'

COMPUTE IF: NOT (QAccomdat.SubLet = Yes)

Im := 'In'

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

ASK ALWAYS:

Premium

SHOW CARD Q

Do ^you have any insurance policies which cover you for any of the things shown on this card (these are not life/death policies)?

PLEASE INCLUDE ANY INSURANCE PROVIDED BY AN EMPLOYER OR A PENSION SCHEME.

IF ANY POLICY WAS DEALT WITH EARLIER (AT MORTGAGE SECTION), DO NOT REPEAT HERE.

(1) Yes
(2) No

COMPUTE ALWAYS:

```
QAccomdat.Premium := Premium
```

FRS0405A.QPolicies

```
COMPUTE IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
```

`Policy[index].InsSeq := index`

FRS0405A.QPolicies.Policy[]

```
RECORD IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
```

InsSeq

QPolicies

Sequence number

1..6

```
COMPUTE IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
```

next := 'first'

```
COMPUTE IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: InsSeq > 1
```

next := 'next'

```
ASK IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
```

NumPols

QPolicies

Friendly society policies for sickness include:

Benevolent fund (unless stated to be a charity)
Burial club
Beneden Healthcare Society (formerly post Office and Civil Service Sanatorium Society)
Death levy
Family Service Unit
Fireman's benevolent fund
Hospital savings association (HSA)
Hospital Saturday Fund
Medical aid
Mutual Aid
Oddfellows

SET [9] OF
(1) Personal accident insurance
(2) Private medical
(3) Permanent health insurance
(4) Critical illness cover
(5) Friendly society sickness benefit
(6) To provide an income while in hospital
(7) Nursing home/long-term care
(8) Any other sickness insurance
(9) Unemployment/Redundancy

```
WARN IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     RESERVECHECK
```

RESERVECHECK

```
ASK IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: (((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN
     NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN
     NumPols)
```

PolPay

QPolicies

Who pays the premiums?

(1) The person(s) insured
(2) Someone else
(3) Both of the above

```
ASK IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: (((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN
     NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN
     NumPols)
     AND: (PolPay = Selfins) OR (PolPay = Both)
```

PolAmt

QPolicies

If the premium is paid by both the person insured and someone else, enter the part paid by the respondent only.

0.00..9997.00

```
COMPUTE IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: (((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN
     NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN
     NumPols)
     AND: (PolPay = Selfins) OR (PolPay = Both)
     AND: PolAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: (((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN
     NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN
     NumPols)
     AND: (PolPay = Selfins) OR (PolPay = Both)
     AND: PolAmt > 0
```

PolPx

QPolicies

^Pd97Txt

OPEN

ASK IF: *Premium = Yes*
AND: *In loop FOR index := 1 TO 6*
AND: *(index = 1) OR (Policy[index - 1].PolMore = Yes)*
AND: *(((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN NumPols)*
AND: *(PolPay = Selfins) OR (PolPay = Both)*
AND: *PolAmt > 0*

PolPd

QPolicies

How long did that cover?

(1)	One week
(2)	Two weeks
(3)	Three weeks
(4)	Four weeks
(5)	Calendar month
(7)	Two Calendar months
(8)	Eight times a year
(9)	Nine times a year
(10)	Ten times a year
(13)	Three months/13 weeks
(26)	Six months/26 weeks
(52)	One Year/12 months/52 weeks
(90)	Less than one week
(95)	One off/lump sum
(97)	None of these (EXPLAIN IN A NOTE)

ASK IF: *Premium = Yes*
AND: *In loop FOR index := 1 TO 6*
AND: *(index = 1) OR (Policy[index - 1].PolMore = Yes)*
AND: *(((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN NumPols)*
AND: *(PolPay = Selfins) OR (PolPay = Both)*
AND: *PolAmt > 0*
AND: *PolPd = Note*

PolPx

QPolicies

^Pd97Txt

OPEN

```
WARN IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: (((((PerAcc IN NumPols) OR (Health IN NumPols)) OR (Crit IN
     NumPols)) OR (Sick IN NumPols)) OR (Hosp IN NumPols)) OR (Redun IN
     NumPols)
     AND: (PolPay = Selfins) OR (PolPay = Both)
     AND: PolAmt > 0
     AND: Edit = Yes
     PolPd <> Note
```

EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.

```
ASK IF: Premium = Yes
     AND: In loop FOR index := 1 TO 6
     AND: (index = 1) OR (Policy[index - 1].PolMore = Yes)
     AND: InsSeq < 6
```

PolMore

QPolicies

SHOW CARD Q

Do ^you have any more policies for any of the things shown on this card?

(1) Yes
(2) No

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN IF: *Premium = Yes*
RESERVECHECK

RESERVECHECK

WARN IF: *Premium = Yes*
RESERVECHECK

RESERVECHECK

WARN IF: *Premium = Yes*
RESERVECHECK

RESERVECHECK

WARN IF: *Premium = Yes*
RESERVECHECK

RESERVECHECK

FRS0405A.QTeleV

Televisions

COMPUTE ALWAYS:

```
colour := 'colour'
```

ASK ALWAYS:

ConTV

QTeleV

Does your household have any of the following items?

...a ^colour TV set?

INCLUDE ITEMS STORED BUT IN WORKING ORDER, AND ITEMS UNDER REPAIR.

(1) One only
(2) more than one
(3) none

COMPUTE ALWAYS:

```
colour := 'black and white'
```

ASK ALWAYS:

ConTV

QTeleV

Does your household have any of the following items?

...a ^colour TV set?

INCLUDE ITEMS STORED BUT IN WORKING ORDER, AND ITEMS UNDER REPAIR.

(1) One only
(2) more than one
(3) none

ASK IF: (Over75 > 0) AND ((ConTV[1] IN [One .. More]) OR (ConTV[2] IN [One .. More]))

TVLic

QTeleV

Do you claim a concessionary television licence?

INTERVIEWER: THESE ARE FREE TV LICENCES FOR THOSE AGED 75 OR OVER.

(1) Yes
(2) No

Warn always:

```
RESERVECHECK
```

RESERVECHECK

Warn always:

```
RESERVECHECK
```

RESERVECHECK

Warn always:

```
RESERVECHECK
```

RESERVECHECK

Warn always:

```
RESERVECHECK
```

RESERVECHECK

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

COMPUTE ALWAYS:

```
PersList[1] := ''
```

COMPUTE ALWAYS:

```
PersList[2] := ''
```

COMPUTE ALWAYS:

```
PersList[3] := ''
```

COMPUTE ALWAYS:

```
Elig[1] := 0
```

COMPUTE ALWAYS:

```
Elig[2] := 0
```

COMPUTE ALWAYS:

```
Elig[3] := 0
```

COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize
AND: ((DMAge[Loop1] IN [16 .. 50]) AND (PRec[Loop1].Sex = Female)) OR (DMAge[Loop1] IN [0 .. 15])

```
PersList[1] := (PersList[1] + STR(Loop1,2) + ' : ' +
DMName[Loop1] + '
')
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize
     AND: ((DMAge[Loop1] IN [16 .. 50]) AND (PRec[Loop1].Sex = Female)) OR
     (DMAge[Loop1] IN [0 .. 15])
```

```
Elig[1] := (Elig[1] + 1)
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].TypeEd IN [Nursery .. Nonadv]
```

```
PersList[2] := (PersList[2] + STR(Loop1,2,0) + ' : ' +
DMName[Loop1] + '
')
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].TypeEd IN [Nursery .. Nonadv]
```

```
Elig[2] := (Elig[2] + 1)
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize
     AND: (DMAge[Loop1] IN [2 .. 18]) AND (PRec[Loop1].TypeEd IN [Nursery ..
     Nonadv])
```

```
PersList[3] := (PersList[3] + STR(Loop1,2,0) + ' : ' +
DMName[Loop1] + '
')
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize
     AND: (DMAge[Loop1] IN [2 .. 18]) AND (PRec[Loop1].TypeEd IN [Nursery ..
     Nonadv])
```

```
Elig[3] := (Elig[3] + 1)
```

FRS0405A.QWelfare

Questions about free meals etc

COMPUTE IF: *PAllCh > 0*

incl_child := ' (including any of your children under 16)'

COMPUTE IF: *PAllCh = 1*

incl_child := ' (including your child under 16)'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 1*

READ_OUT := 'READ OUT: PROMPT AT EACH ITEM INDIVIDUALLY.'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *Elig[1] >= 1*

WelfMilk := '...any free welfare milk?'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *NOT (Elig[1] >= 1)*

WelfMilk := '(not used)'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *Elig[2] >= 1*

SchMilk := '...any free school milk?'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *NOT (Elig[2] >= 1)*

SchMilk := '(not used)'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *Elig[3] >= 1*

SchMeal := '...any free school meals?'

COMPUTE IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *NOT (Elig[3] >= 1)*

SchMeal := '(not used)'

ASK IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*

FreeItem

QWelfare

QUESTIONS ABOUT FREE SCHOOL MEALS AND WELFARE MILK.

In the last 7 days, have ^you^incl_child had...

^READ_OUT

SET [3] OF
(1) ^WelfMilk
(2) ^SchMilk
(3) ^SchMeal
(4) None of these

CHECK IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *None IN FreeItem*
FreeItem.CARDINAL = 1

'None of these' is an exclusive code for this question.

CHECK IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *WMilk IN FreeItem*
Elig[1 > 0

Code 1 is not valid for this question.

CHECK IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *SMilk IN FreeItem*
Elig[2 > 0

Code 2 is not valid for this question.

CHECK IF: *((Elig[1] + Elig[2]) + Elig[3]) > 0*
AND: *SMeal IN FreeItem*
Elig[3 > 0

Code 3 is not valid for this question.

FRS0405A.QWelfare.WMkQ[]

```
RECORD IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

Person

QWelfare

Person identifier.

0..14

```
RECORD IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

BenUnit

QWelfare

BU number of recipient.

0..7

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: PHHSize = 1
```

WMkPer := 1

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: PHHSize = 1
```

NameOf := 'you'

```
ASK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (PHHSize = 1)
```

WMkPer

QWelfare

Who received the free welfare milk?

INTERVIEWER TYPE IN PERSON NUMBER.

^PersList[1]

0..14

```
CHECK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (PHHSize = 1)
     AND: WMkPer = RESPONSE
     (WMkPer > 0) AND (WMkPer <= PHHSize)
```

This code is not valid for this question.

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (PHHSize = 1)
```

NameOf := DMName[[WMkPer]

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

Person := WMkPer

```
ASK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: WMkPer = RESPONSE
```

WMkIt

QWelfare

Thinking just of the PAST SEVEN DAYS ending yesterday - how many pints did ^NameOf receive?

0..97

```
WARN IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: WMkPer = RESPONSE
     AND: Edit = No
     WMkIt <= 7
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: Elig[1] > 1
```

WMIntro

QWelfare

INTERVIEWER PROMPT: Has anyone else had any free welfare milk during the past seven days ending yesterday?

(1) Yes
(2) No

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (Elig[1] > 1)
```

WMIntro := No

FRS0405A.QWelfare (continued)

Questions about free meals etc

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

WMkQ[Index1].BenUnit := DMBU[[WMkQ[Index1].WMkPer]

```
CHECK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: In loop FOR Index2 := 1 TO 5
     WMkQ[Index2].WMkPer <> WMkQ[Index1].WMkPer
```

You have already entered this person number.

FRS0405A.QWelfare.SMkQ[]

```
RECORD IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

Person

QWelfare

Person identifier.

0..14

```
RECORD IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

BenUnit

QWelfare

BU number of recipient.

0..7

```
ASK IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

SMkPer

QWelfare

Free School Milk may be available if the child is in full-time education at a state school. The rules for offering Free School Milk differ by Local Education Authority. The age of children for whom it can be claimed also varies by LEA.

0..14

```
COMPUTE IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

```
Person := SMkPer
```

ASK IF: *SMilk IN FreeItem*
  AND: *In loop FOR Index1 := 1 TO 5*
  AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*

SMkIt

QWelfare

Thinking just of the PAST SEVEN DAYS ending yesterday - how many cartons or bottles did ^DMName[SmkPer] receive?

0..97

WARN IF: *SMilk IN FreeItem*
  AND: *In loop FOR Index1 := 1 TO 5*
  AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*
  AND: *Edit = No*
  SMkIt <= 6

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

ASK IF: *SMilk IN FreeItem*
  AND: *In loop FOR Index1 := 1 TO 5*
  AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*
  AND: *Elig[2] > 1*

SMIntro

QWelfare

INTERVIEWER PROMPT: Has any other child had any free school milk during the past seven days ending yesterday?

ONLY APPLICABLE TO CHILDREN AT STATE SCHOOLS.

(1) Yes
(2) No

COMPUTE IF: *SMilk IN FreeItem*
  AND: *In loop FOR Index1 := 1 TO 5*
  AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*
  AND: *NOT (Elig[2] > 1)*

SMIntro := No

FRS0405A.QWelfare (continued)

Questions about free meals etc

```
COMPUTE IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

SMkQ[Index1].BenUnit := DMBU[[SMkQ[Index1].SMkPer]

```
CHECK IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: In loop FOR Index2 := 1 TO 5
     SMkQ[Index2].SMkPer <> SMkQ[Index1].SMkPer
```

You have already entered this person number.

FRS0405A.QWelfare.SMlQ[]

```
RECORD IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

BenUnit

QWelfare

BU number of recipient.

0..7

```
RECORD IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

Person

QWelfare

Person identifier.

0..14

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: PHHSize = 1
```

MLPer := 1

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: PHHSize = 1
```

have_you := 'have you'

***ASK IF:** SMeal IN FreeItem*
    ***AND:** In loop FOR Index1 := 1 TO 5*
    ***AND:** (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)*
    ***AND:** NOT (PHHSize = 1)*

MLPer

QWelfare

Who received the free school meals?

ONLY APPLICABLE TO CHILDREN AT STATE SCHOOLS. CAN INCLUDE 16-18 YEAR OLDS. INTERVIEWER TYPE IN PERSON NUMBER.

^PersList[3]

0..14

***COMPUTE IF:** SMeal IN FreeItem*
    ***AND:** In loop FOR Index1 := 1 TO 5*
    ***AND:** (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)*
    ***AND:** NOT (PHHSize = 1)*

have_you := ('has ' + DMName[[MLPer])

***COMPUTE IF:** SMeal IN FreeItem*
    ***AND:** In loop FOR Index1 := 1 TO 5*
    ***AND:** (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)*

Person := MLPer

***ASK IF:** SMeal IN FreeItem*
    ***AND:** In loop FOR Index1 := 1 TO 5*
    ***AND:** (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)*
    ***AND:** MLPer = RESPONSE*

SMlIt

QWelfare

Thinking just of the PAST SEVEN DAYS ending yesterday, how many free school meals ^have_you had?

0..97

***WARN IF:** SMeal IN FreeItem*
    ***AND:** In loop FOR Index1 := 1 TO 5*
    ***AND:** (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)*
    ***AND:** MLPer = RESPONSE*
    ***AND:** Edit = No*
    SMlIt <= 21

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: Elig[3] > 1
```

MLIntro

QWelfare

INTERVIEWER PROMPT: Has any other child had any free school meals during the past seven days ending yesterday?

ONLY APPLICABLE TO CHILDREN AT STATE SCHOOLS. CAN INCLUDE 16-18 YEAR OLDS.

(1) Yes
(2) No

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: NOT (Elig[3] > 1)
```

MLIntro := No

FRS0405A.QWelfare (continued)

Questions about free meals etc

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

SMlQ[Index1].BenUnit := DMBU[[SMlQ[Index1].MLPer]

```
CHECK IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: In loop FOR Index2 := 1 TO 5
     SMlQ[Index2].MLPer <> SMlQ[Index1].MLPer
```

You have already entered this person number.

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.WMkQ[Loop1].WMkPer = RESPONSE

Loop5 := QWelfare.WMkQ[Loop1].WMkPer
```

```
CHECK IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.WMkQ[Loop1].WMkPer = RESPONSE
     ((Loop5 > 0) AND (Loop5 <= HHSize)) AND (((IN(DMAge[Loop5],[16..50]))
     AND (PRec[Loop5].Sex = Female)) OR (IN(DMAge[Loop5],[0..15])))
```

This code is not valid for this question.

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.WMkQ[Loop1].WMkPer = RESPONSE

QWelfare.WMkQ[Loop1].BenUnit := DMBU[Loop5]
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMkQ[Loop1].SMkPer = RESPONSE

Loop5 := QWelfare.SMkQ[Loop1].SMkPer
```

```
CHECK IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMkQ[Loop1].SMkPer = RESPONSE
     ((Loop5 > 0) AND (Loop5 <= HHSize)) AND (IN(PRec[Loop5].TypeEd,[???]))
```

This code is not valid for this question.

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMkQ[Loop1].SMkPer = RESPONSE

QWelfare.SMkQ[Loop1].BenUnit := DMBU[Loop5]
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SM1Q[Loop1].MLPer = RESPONSE

Loop5 := QWelfare.SM1Q[Loop1].MLPer
```

```
CHECK IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SM1Q[Loop1].MLPer = RESPONSE
     (((Loop5 > 0) AND (Loop5 <= HHSize)) AND (IN(DMAge[Loop5],[2..18])))
     AND (IN(PRec[Loop5].TypeEd,[???]))
```

This code is not valid for this question.

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SM1Q[Loop1].MLPer = RESPONSE

QWelfare.SM1Q[Loop1].BenUnit := DMBU[Loop5]
```

WARN IF: *In loop FOR Loop1 := 1 TO 5*
AND: *PRec[QWelfare.SMlQ[Loop1].MLPer].TypeEd IN [Nursery, Primry, MidPri .. Nonadv]*
(IN(QWelfare.SMlQ[Loop1].SMlIt,[0..5])) AND INVOLVING(QWelfare.SMlQ[Loop1].SMlIt)

That's ^QWelfare.SMlQ[Loop1].SMlIt meals - for this type of school the weekly maximum would normally be 5 (ie. one meal per day) - please check.

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

COMPUTE IF: *SMeal IN QWelfare.FreeItem*

QAccomdat.SchMeal := Yes

COMPUTE IF: *NOT (SMeal IN QWelfare.FreeItem)*

QAccomdat.SchMeal := No

COMPUTE IF: *SMilk IN QWelfare.FreeItem*

QAccomdat.SchMilk := Yes

COMPUTE IF: *NOT (SMilk IN QWelfare.FreeItem)*

QAccomdat.SchMilk := No

COMPUTE IF: *WMilk IN QWelfare.FreeItem*

QAccomdat.WelfMilk := Yes

COMPUTE IF: *NOT (WMilk IN QWelfare.FreeItem)*

QAccomdat.WelfMilk := No

FRS0405A.QChCare

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Child[Index2].BenUnit := DMBU[[Index2]

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Child[Index2].Person := Index2

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] = 1
```

you := 'you'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] = 1
```

PNames := DMName[[DMParent1[[Index2]]

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] > 1
```

you := 'either of you'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] > 1
```

PNames := (DMName[[DMParent1[[Index2]] + ' AND ' + DMName[[DMParent2[[Index2]])

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

ChName := DMName[[Index2]

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

ChAge := DMAge[[Index2]

FRS0405A.QChCare.Child[]

```
RECORD IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

BenUnit

QChCare

BU number of person

0..7

```
RECORD IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Person

QChCare

Person identifier.

0..14

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: NatCen = NI
```

HBenCTRT := 'Housing Benefit/rent/rates rebate'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: NOT (NatCen = NI)
```

HBenCTRT := 'Housing Benefit/Council Tax Benefit'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Disp

QChCare

The next questions are about childcare facilities for ^ChName.

(THE QUESTIONS SHOULD BE DIRECTED AT ^PNames WHEREVER POSSIBLE)

(1) Press <Enter> to continue.

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ParentNo = 1
```

LoneParent := Yes

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ParentNo = 1
```

NCDVLP := Yes

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: NOT (ParentNo = 1)
```

LoneParent := No

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Care

QChCare

Childcare refers to looking after a child on a regular basis, i.e. an ongoing arrangement to provide care for a child in the absence of the parent or parents. The type of care that would be excluded here would be when a single one-off arrangement is made. Childcare can be provided by anyone outside of the actual parents.

(1) Yes
(2) No

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
```

Cost

QChCare

Does your child-care for ^ChName cost you anything?

INTERVIEWER: DO NOT INCLUDE CERTIFICATE OF ELIGIBILTY.

(1) Yes
(2) No

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
```

ChLook

QChCare

SHOWCARD R
Who looks after ^ChName?
PROBE: Anyone else? CODE ALL THAT APPLY.

INTERVIEWER: CLOSE RELATIVE = Respondent's PARTNER, PARENT (inc. STEP-), SON or DAUGHTER (inc. STEP-), BROTHER or SISTER, or SPOUSE of any of these.

SET [10] OF
(1) Close relative
(2) Other relative
(3) Friend/Neighbour
(4) Childminder
(5) Nursery/School/Playgroup
(6) Creche
(7) Employer provided nursery
(8) Nanny/Au pair
(9) Before school/Breakfast club/After School or holiday Play Scheme
(10) Other

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
```

ChFar

QChCare

How long does it take to travel from your home to the place where ^ChName is looked after?

IF TWO OR MORE TYPES OF CARE, RECORD TRAVEL TIME FOR PLACE OF CARE WITH THE MOST HOURS PER WEEK.

(1) Cared for at home
(2) Less than half an hour
(3) Half to one hour
(4) More than one, but less than 2 hours
(5) 2 hours or more

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: ChMind IN ChLook
```

childminder := 'Childminder'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: ChMind IN ChLook
```

Registrd

QChCare

Can I just check, is the ^childminder registered, or not?

(1) Registered
(2) Not registered

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: ChMind IN ChLook
     AND: Registrd[1] = Registered
```

ChText := childminder

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nursery IN ChLook
```

childminder := 'Nursery / Playgroup'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nursery IN ChLook
```

Registrd

QChCare

Can I just check, is the ^childminder registered, or not?

(1) Registered
(2) Not registered

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nursery IN ChLook
     AND: Registrd[2] = Registered
     AND: ChText =
```

ChText := childminder

```
COMPUTE IF: AllCh > 0
      AND: In loop FOR Index2 := 1 TO HHSize
      AND: DMAge[Index2] IN [0 .. 15]
      AND: Care = Yes
      AND: Nursery IN ChLook
      AND: Registrd[2] = Registered
      AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + childminder)

```
COMPUTE IF: AllCh > 0
      AND: In loop FOR Index2 := 1 TO HHSize
      AND: DMAge[Index2] IN [0 .. 15]
      AND: Care = Yes
      AND: Creche IN ChLook
```

childminder := 'Creche'

```
ASK IF: AllCh > 0
      AND: In loop FOR Index2 := 1 TO HHSize
      AND: DMAge[Index2] IN [0 .. 15]
      AND: Care = Yes
      AND: Creche IN ChLook
```

Registrd

QChCare

Can I just check, is the ^childminder registered, or not?

(1) Registered
(2) Not registered

```
COMPUTE IF: AllCh > 0
      AND: In loop FOR Index2 := 1 TO HHSize
      AND: DMAge[Index2] IN [0 .. 15]
      AND: Care = Yes
      AND: Creche IN ChLook
      AND: Registrd[3] = Registered
      AND: ChText =
```

ChText := childminder

```
COMPUTE IF: AllCh > 0
      AND: In loop FOR Index2 := 1 TO HHSize
      AND: DMAge[Index2] IN [0 .. 15]
      AND: Care = Yes
      AND: Creche IN ChLook
      AND: Registrd[3] = Registered
      AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + childminder)

```
COMPUTE IF: AllCh > 0
      AND: In loop FOR Index2 := 1 TO HHSize
      AND: DMAge[Index2] IN [0 .. 15]
      AND: Care = Yes
      AND: EmpNurs IN ChLook
```

childminder := 'Employer Provided Nursery'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: EmpNurs IN ChLook
```

Registrd

QChCare

Can I just check, is the ^childminder registered, or not?

(1) Registered
(2) Not registered

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: EmpNurs IN ChLook
     AND: Registrd[4] = Registered
     AND: ChText =
```

ChText := childminder

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: EmpNurs IN ChLook
     AND: Registrd[4] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + childminder)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nanny IN ChLook
```

childminder := 'Nanny / Aupair'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nanny IN ChLook
```

Registrd

QChCare

Can I just check, is the ^childminder registered, or not?

(1) Registered
(2) Not registered

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nanny IN ChLook
     AND: Registrd[5] = Registered
     AND: ChText =
```

ChText := childminder

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Nanny IN ChLook
     AND: Registrd[5] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + childminder)

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: ((((((Registrd[1] = Registered) OR (Registrd[2] = Registered)) OR
     (Registrd[3] = Registered)) OR (Registrd[4] = Registered)) OR
     (Registrd[5] = Registered)) AND (ChAge < 13)) AND
     ((((((QRenting.HBenAmt > 0) OR (QRenting.HBenefit = Yes)) OR
     (QCounTax.CTRebAmt > 0)) OR (QCounTax.CTReb = Yes)) OR
     (QNIRates.RTRebAmt > 0)) OR (QNIRates.RTReb = Yes))
```

BenCCDis

QChCare

You said earlier that you get (^HBenCTRT). Does the benefit take account of the cost of the ^ChText?

(1) Yes
(2) No

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
```

ChHr1

QChCare

INTERVIEWER: Unpaid childcare is any care provided on a regular basis for which no cost is attached. This may include regular after school care provided by grandparents, a neighbour etc.

0..60

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Edit = No
     ChHr1 < 55
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
```

ChHr2

QChCare

INTERVIEWER: Unpaid childcare is any care provided on a regular basis for which no cost is attached. This may include regular after school care provided by grandparents, a neighbour etc.

0..60

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Edit = No
     ChHr2 < 69
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: ChHr1 > 0
```

ChAmt1

QChCare

How much does it usually cost you per week for ^ChName ...READ OUT:

i) ...in term time?

IF UNABLE TO ATTRIBUTE COSTS PER CHILD, THEN ENTER AN ESTIMATE BY DIVIDING TOTAL CHILDCARE COSTS BY NUMBER OF CHILDREN.
IF NO DIFFERENCE BETWEEN TERM TIME AND HOLIDAYS, ENTER SAME FIGURE AT BOTH QUESTIONS.

0.00..400.00

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: ChHr1 > 0
     AND: ChAmt1 = NONRESPONSE
```

```
HMissVar := (HMissVar + 1)
```

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: ChHr1 > 0
     AND: Edit = No
     ChAmt1 < 130
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: ChHr2 > 0
```

ChAmt2

QChCare

How much does it usually cost you per week for ^ChName ...READ OUT:

ii) ...in the school holidays?

IF UNABLE TO ATTRIBUTE COSTS PER CHILD, THEN ENTER AN ESTIMATE BY DIVIDING TOTAL CHILDCARE COSTS BY NUMBER OF CHILDREN.
IF NO DIFFERENCE BETWEEN TERM TIME AND HOLIDAYS, ENTER SAME FIGURE AT BOTH QUESTIONS.

0.00..400.00

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: ChHr2 > 0
     AND: ChAmt2 = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: ChHr2 > 0
     AND: Edit = No
     ChAmt2 < 140
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
```

ChPay

QChCare

Do you make any other payment in kind, or other form of compensation?

SET [3] OF
(1) Payment in kind
(2) Exchange basis
(3) Other
(4) No (other) cost or payment of any kind

```
CHECK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: Cost = Yes
     AND: NoCost IN ChPay
     ChPay.CARDINAL = 1
```

Code 3 is exclusive for this question.

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: LoneParent = Yes
```

CLone

QChCare

Does anyone else pay for ^ChName to have childcare?

(1) Yes
(2) No

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: LoneParent = Yes
     AND: (CLone = Yes) AND (ChHr1 > 0)
```

ChAmt3

QChCare

How much do others usually pay per week for ^ChName ...READ OUT:

i) ...to have childcare in term time?

IF UNABLE TO ATTRIBUTE COSTS PER CHILD, THEN ENTER AN ESTIMATE BY DIVIDING TOTAL CHILDCARE COSTS PAID BY OTHERS BY NUMBER OF CHILDREN.
IF NO DIFFERENCE BETWEEN TERM TIME AND HOLIDAYS, ENTER SAME FIGURE AT BOTH QUESTIONS.

0.00..400.00

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: LoneParent = Yes
     AND: (CLone = Yes) AND (ChHr1 > 0)
     AND: Edit = No
     ChAmt3 < 130
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: LoneParent = Yes
     AND: (CLone = Yes) AND (ChHr2 > 0)
```

ChAmt4

QChCare

How much do others usually pay per week for ^ChName ...READ OUT:

ii) ...to have childcare in the school holidays?

IF UNABLE TO ATTRIBUTE COSTS PER CHILD, THEN ENTER AN ESTIMATE BY DIVIDING TOTAL CHILDCARE COSTS PAID BY OTHERS BY NUMBER OF CHILDREN.
IF NO DIFFERENCE BETWEEN TERM TIME AND HOLIDAYS, ENTER SAME FIGURE AT BOTH QUESTIONS.

0.00..400.00

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Care = Yes
     AND: LoneParent = Yes
     AND: (CLone = Yes) AND (ChHr2 > 0)
     AND: Edit = No
     ChAmt4 < 140
```

Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     RESERVECHECK
```

RESERVECHECK

FRS0405A.QChCare (continued)

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Edit = No
     AND: ((DMBU[Index2] = 1) AND (Child[Index2].BenCCDis = Yes)) AND
     ((ChMind IN Child[Index2].ChLook) OR (Nursery IN
     Child[Index2].ChLook))
     (Child[Index2].Registrd[1] = Registered) OR (Child[Index2].Registrd[2]
     = Registered)
```

Please check this answer with respondent: earlier they said they got the 'Child Care Disregard' for the benefit(s) shown below, and this is usually only possible if the childminder/nursery IS REGISTERED.

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *In loop FOR Loop1 := 1 TO 14*

```
     AND: ((((QChCare.Child[Loop1].Registrd[1] = Registered) OR
     (QChCare.Child[Loop1].Registrd[2] = Registered)) OR
     (QChCare.Child[Loop1].Registrd[3] = Registered)) OR
     (QChCare.Child[Loop1].Registrd[4] = Registered)) OR
     (QChCare.Child[Loop1].Registrd[5] = Registered)
```

ChRegis := (ChRegis + 1)

FRS0405A.QCare

Questions about carers/cared for

Ask always:

NeedHelp

QCare

In some households, there are people who receive help or support because they have long-term physical or mental ill-health or disability (or problems relating to old age).

SHOW CARD S
Is there anyone in this household who receives any of these kinds of help or looking after?

INTERVIEWER: INCLUDE HELP FROM WIFE/HUSBAND/PARTNER/OTHER FAMILY MEMBER.

(1) Yes
(2) No

Ask always:

GiveHelp

QCare

SHOW CARD S
And how about people not living with you: do you (or does anyone else in this household) provide any help or support for anyone not living with you who has a long-term physical or mental ill-health problem or disability, or problems relating to old age?

EXCLUDE HELP GIVEN AS PART OF A PERSON'S PAID JOB, EG. IF RESPONDENT WORKS FOR SOCIAL SERVICES.

(1) Yes
(2) No

FRS0405A.QCare.QRecHelp

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: NeedHelp = Yes
     AND: In loop FOR Idy := 1 TO HHSize
```

LName[Idy] := DMName[[Idy]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[15] := 'Parent outside household'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[16] := 'Other Parent outside household'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[17] := 'Child outside household'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[18] := 'Spouse outside household'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[19] := 'Other relative'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[20] := 'Friend/neighbour'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[21] := 'Client of voluntary organisation'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: GiveHelp = Yes
```

LName[22] := 'Other non-household'

Ask if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*

QNeedPer

QCare

If they provide help or give help for MORE THAN ONE individual in any one category of:

15: Parent outside household
16: Other parent outside household
18: Child outside household
19: Relative
20: Friend/neighbour
21: Client of voluntary organisation
22: Other non-household

Please make a Note here <Ctrl+M> to tell us how many more people are involved.

SET [5] OF

(1)	^LName[1]
(2)	^LName[2]
(3)	^LName[3]
(4)	^LName[4]
(5)	^LName[5]
(6)	^LName[6]
(7)	^LName[7]
(8)	^LName[8]
(9)	^LName[9]
(10)	^LName[10]
(11)	^LName[11]
(12)	^LName[12]
(13)	^LName[13]
(14)	^LName[14]
(15)	^LName[15]
(16)	^LName[16]
(17)	^LName[17]
(18)	^LName[18]
(19)	^LName[19]
(20)	^LName[20]
(21)	^LName[21]
(22)	^LName[22]

Ask if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *Per22 IN QNeedPer*

NeedPerO

QCare

Who is the other person outside the household receiving help or being looked after?

STRING[40]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] IN [Per1 .. Per22]

NeedNum := ORD(QNeedPer[Idx])
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NeedNum IN [1 .. 14]

NeedName := DMName[[NeedNum]
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per15

NeedName := 'the PARENT'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per16

NeedName := 'the OTHER PARENT'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per17

NeedName := 'the CHILD'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per18

NeedName := 'the SPOUSE'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per19

NeedName := 'the RELATIVE'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per20

NeedName := 'the FRIEND/NEIGHBOUR'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per21

NeedName := 'the CLIENT of a voluntary organisation'
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per22
```

NeedName := UPCASE(NeedPerO)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
```

Recip[Idx].NeedPer := NeedNum

FRS0405A.QCare.QRecHelp.Recip[]

```
RECORD IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
```

NeedPer

QCare

Who is receiving help/being looked after.

0..22

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
```

LNeedPer := NeedPer

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
```

LNeedName := PNeedName

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
```

Freq

QCare

How frequently does ^LNeedName receive such help?

(1) Continuously
(2) Several times a day
(3) Once or twice a day
(4) Several times a week
(5) Once a week
(6) Less frequently

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously, SevDay]
```

DayNight

QCare

And does ^LNeedName receive help during the daytime; or at night; or both in the day and at night?

(1) Daytime only
(2) At night only
(3) Both day and night

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

`Relatives := ''`

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

`Friend := ''`

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

`LAHelp := ''`

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

`Domestic := ''`

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

`Nurse := ''`

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

`Helper := ''`

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
```

Relatives := 'Relatives'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
```

Friend := 'Friends/Neighbours'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
     AND: NatCen = NI
```

LAHelp := 'Social Services home help or home care worker'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
     AND: NOT (NatCen = NI)
```

LAHelp := 'Local Authority home help or home care worker'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
```

Domestic := 'Private domestic help'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
```

Nurse := 'District nurse, health visitor or other kind of nurse'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: NOT (LNeedPer > 14)
```

Helper := 'Other outside helpers'

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
```

WhoLook

QCare

Who looks after, or provides help for ^LNeedName? Anyone else?

CODE ALL THAT APPLY.

SET [5] OF
(1) ^DMName[1]
(2) ^DMName[2]
(3) ^DMName[3]
(4) ^DMName[4]
(5) ^DMName[5]
(6) ^DMName[6]
(7) ^DMName[7]
(8) ^DMName[8]
(9) ^DMName[9]
(10) ^DMName[10]
(11) ^DMName[11]
(12) ^DMName[12]
(13) ^DMName[13]
(14) ^DMName[14]
(15) ^Relatives
(16) ^Friend
(17) ^LAHelp
(18) ^Domestic
(19) ^Nurse
(20) ^Helper

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: LNeedPer <= 14
     NOT(IN(LNeedPer,WhoLook))
```

You've included ^LNeedName as looking after him-/herself. Please remove him/her from the answer at WhoLook.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 14
     AND: Count IN WhoLook
     DMAge[[Count] > 0
```

Code ^Count is not valid for this question.

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Relative
```

HelpArr := 'es the relative'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Relative
```

HCount := 15

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Friends
```

HelpArr := 'es the friend'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Friends
```

HCount := 16

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NatCen = NI
```

HelpArr := ('es the Social Services home help or ' + 'home care worker')

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NOT (NatCen = NI)
```

HelpArr := 'es the LA home help or home care worker'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
```

HCount := 17

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Domestic
```

HelpArr := 'es the private domestic help'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Domestic
```

HCount := 18

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Nurse
```

HelpArr := 'es the nurse'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Nurse
```

HCount := 19

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Helpers
```

HelpArr := 'es the outside helper'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Helpers
```

HCount := 20

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per1
```

HelpArr := ('es ' + DMName[1)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per1
```

HCount := 1

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per2
```

HelpArr := ('es ' + DMName[2)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per2
```

HCount := 2

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per3
```

HelpArr := ('es ' + DMName[3)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per3
```

HCount := 3

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per4
```

HelpArr := ('es ' + DMName[4)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per4
```

HCount := 4

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per5
```

HelpArr := ('es ' + DMName[5)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per5
```

HCount := 5

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per6
```

HelpArr := ('es ' + DMName[6)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per6
```

```
HCount := 6
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per7
```

```
HelpArr := ('es ' + DMName[7)
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per7
```

```
HCount := 7
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per8
```

```
HelpArr := ('es ' + DMName[8)
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per8
```

```
HCount := 8
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per9
```

```
HelpArr := ('es ' + DMName[9)
```

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per9
```

HCount := 9

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per10
```

HelpArr := ('es ' + DMName[10)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per10
```

HCount := 10

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per11
```

HelpArr := ('es ' + DMName[11)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per11
```

HCount := 11

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per12
```

HelpArr := ('es ' + DMName[12)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per12
```

HCount := 12

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per13
```

HelpArr := ('es ' + DMName[13)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per13
```

HCount := 13

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per14
```

HelpArr := ('es ' + DMName[14)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per14
```

HCount := 14

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
```

Hour

QCare

SHOW CARD T
About how many hours a week, on average, do^HelpArr spend actually providing help for or looking after ^LNeedName?

(1) 0-4 hours per week
(2) 5-9 hours per week
(3) 10-19 hours per week
(4) 20-34 hours per week
(5) 35-49 hours per week
(6) 50-99 hours per week
(7) 100 or more hours per week
(8) Varies - under 20 hours per week
(9) Varies - 20-34 hours per week
(10) Varies - 35 hours a week or more

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously .. OWeek]
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: Edit <> Yes
     AND: HCount <= 14
     AND: DMAge[HCount] IN [1 .. 15]
     Hour[Count] <> over100
```

Are you sure that the child(ren) are looking after ^LNeedName for 100 hours a week or more?
If so, suppress warning.

FRS0405A.QCare (continued)

Questions about carers/cared for

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: QRecHelp.QNeedPer <> EMPTY AND (NeedHelp = Yes)
     (((((((((((IN(Per1,QRecHelp.QNeedPer)) OR
     (IN(Per2,QRecHelp.QNeedPer))) OR (IN(Per3,QRecHelp.QNeedPer))) OR
     (IN(Per4,QRecHelp.QNeedPer))) OR (IN(Per5,QRecHelp.QNeedPer))) OR
     (IN(Per6,QRecHelp.QNeedPer))) OR (IN(Per7,QRecHelp.QNeedPer))) OR
     (IN(Per8,QRec
```

Please include the household member who receives regular help, or change 'NeedHelp' to 'No'.

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 1 TO HHSize
     AND: NeedHelp <> EMPTY AND (Index1 IN QRecHelp.QNeedPer)
     NeedHelp = Yes
```

You have coded a household member as receiving regular help, so please change 'NeedHelp' to 'Yes', or remove the household member from 'QNeedPer'.

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: QRecHelp.QNeedPer <> EMPTY AND (GiveHelp = Yes)
     (((((((IN(Per15,QRecHelp.QNeedPer)) OR (IN(Per16,QRecHelp.QNeedPer)))
     OR (IN(Per17,QRecHelp.QNeedPer))) OR (IN(Per18,QRecHelp.QNeedPer))) OR
     (IN(Per19,QRecHelp.QNeedPer))) OR (IN(Per20,QRecHelp.QNeedPer))) OR
     (IN(Per21,QRecHelp.QNeedPer))) OR (IN(Per22,QR
```

Please include the non-household member receiving help from someone in the household, or change 'GiveHelp' to 'No'.

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 15 TO 22
     AND: GiveHelp <> EMPTY AND (Index1 IN QRecHelp.QNeedPer)
     GiveHelp = Yes
```

You have coded a non-household member as receiving help, so please change 'GiveHelp' to 'Yes', or remove the non-household member from 'QNeedPer'.

FRS0405A (continued)

FAMILY RESOURCES SURVEY 2004/2005

CHECK IF: In loop FOR Loop1 := 1 TO 14
AND: Loop1 IN QCare.QRecHelp.QNeedPer
DMAge[Loop1] > 0

Code ^Loop1 is not valid for this question.

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

ASK ALWAYS:

EndDisp

INTERVIEWER:^S4-^S^S END OF 'HOUSEHOLD' SCHEDULE.
NOW ADMINISTER 'BENEFIT UNIT' SCHEDULE(S).
B.U.^S^S MEMBERS
^NameInBU[1]^NameInBU[2]^NameInBU[3]^NameInBU[4]^NameInBU[5]
^NameInBU[6]^NameInBU[7]

TOTAL NUMBER OF BENEFIT UNITS = ^NewBU

PRESS <Ctrl + Enter> TO SELECT BENEFIT UNIT OR TO FILL IN ADMIN DETAILS.
ALTERNATIVELY, PRESS <1> AND <Enter> TO START THE FIRST BENEFIT UNIT.

1..1

RECORD ALWAYS:

HHTime

Time taken from interview start to end of household grid.

Only visible for testing purposes, just press <Enter>.

TIME

RECORD ALWAYS:

HHMins

Total minutes in household grid.

Only visible for testing purposes, just press <Enter>.

0.00..1440.00

COMPUTE IF: *HHTime = EMPTY AND EndDisp <> EMPTY*

HHTime := SYSTIME

COMPUTE IF: *HHTime = EMPTY AND EndDisp <> EMPTY*

HHMins := ((HHTime.ABSTIME - QSignIn.IntSTime.ABSTIME) / 60000)

ASK IF: *Test = Yes*

HHTime

Time taken from interview start to end of household grid.

Only visible for testing purposes, just press <Enter>.

TIME

ASK IF: *Test = Yes*

HHMins

Total minutes in household grid.

Only visible for testing purposes, just press <Enter>.

0.00..1440.00

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

COMPUTE ALWAYS:

NBusRooms := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

AdInBU[1] := 1

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

AdInBU[2] := 1

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child1 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child2 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child3 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child4 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child5 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child6 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child7 := 0

COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child8 := 0

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend = Adult
     AND: AdInBU[1] = EMPTY
```

AdInBU[1] := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend = Adult
     AND: AdInBU[2] = EMPTY
```

AdInBU[2] := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child1 = 0
```

Child1 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child2 = 0
```

Child2 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child3 = 0
```

Child3 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child4 = 0
```

Child4 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child5 = 0
```

Child5 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child6 = 0
```

Child6 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child7 = 0
```

Child7 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child8 = 0
```

Child8 := Loop2