Documentation of Questionnaire/Module 'FRS35' on 25-03-1999 at 16:56

FRS35

FAMILY RESOURCES SURVEY 1998/99

COMPUTE ALWAYS:

Edit := 'No'

COMPUTE ALWAYS:

SCPR := No

COMPUTE ALWAYS:

VerCode := '088_1'

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: *Edit <> Yes*

IVers := ('I_' + VerCode)

COMPUTE IF: *NOT (Edit <> Yes)*

EVers := ('E_' + VerCode)

FRS35.QSerial

Serial number

COMPUTE ALWAYS:

```
KeyString := GETENV('KEYVALUE')
```

COMPUTE ALWAYS:

```
DArea := SUBSTRING(KeyString,1,5)
```

COMPUTE ALWAYS:

```
DAddress := SUBSTRING(KeyString,6,2)
```

COMPUTE ALWAYS:

```
DHhold := SUBSTRING(KeyString,8,1)
```

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 IF: *LEN (DArea) > 0*

```
DArea = STR(Area,5)
```

You've accidentally changed the Area number.
Please change it back to:
Area: ^DArea

CHECK IF: *LEN (DAddress) > 0*
DAddress = STR(Address,2)

You've accidentally changed the Address number.
Please change it back to:
Address: ^DAddress

CHECK IF: *LEN (DHhold) > 0*
DHhold = STR(Hhold,1)

You've accidentally changed the Household number.
Please change it back to:
Hhold: ^DHhold

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

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.Address IN [1 .. 9]*

StrSerial := (STR(QSerial.Area,5) + '0' + STR(QSerial.Address,1))

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

StrSerial := (STR(QSerial.Area,5) + STR(QSerial.Address,2))

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

Serial

Area and Address number

1..9999997

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

Serial := VAL(StrSerial)

FRS35.QDataBag

Sample information

ASK ALWAYS:

Serial

Serial number excluding household number.

1..9999997

ASK ALWAYS:

Hhold

Household number

1..3

ASK ALWAYS:

SurvId

3-letter acronym for survey.

STRING[3]

ASK ALWAYS:

SampYear

Year Code

1998..2005

ASK ALWAYS:

SampMnth

Sample month.

1..12

ASK ALWAYS:

SampQtr

Sample quarter

1..4

ASK ALWAYS:

Attempt

Issue number.

1..7

ASK ALWAYS:

SSTRTReg

Stratifying region: Survey specific.

1..97

ASK ALWAYS:

StaRegGB

Statistical region in GB.

1..12

ASK ALWAYS:

GovRegGB

Government office region in GB.

1..12

ASK ALWAYS:

Country

1..3

ASK ALWAYS:

ACORN

0..97

ASK ALWAYS:

SLA

Local Authority Code.

STRING[4]

ASK ALWAYS:

LAC

Local Authority Code. DSS version

0..997

ASK ALWAYS:

MAFF

If we need it for the Extra addressses.

1..7

ASK ALWAYS:

V2

1..7

ASK ALWAYS:

Padding

To make block compatible with FRSPAF

STRING[66]

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.Hhold := DBData.QDataBag.Hhold

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.SurvId := DBData.QDataBag.SurvId

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.SampYear := DBData.QDataBag.SampYear

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.SampMnth := DBData.QDataBag.SampMnth

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.SampQtr := DBData.QDataBag.SampQtr

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.Attempt := DBData.QDataBag.Attempt

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.SSTRTReg := DBData.QDataBag.SSTRTReg

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.StaRegGB := DBData.QDataBag.StaRegGB

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.GovRegGB := DBData.QDataBag.GovRegGB

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.Country := DBData.QDataBag.Country

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.ACORN := DBData.QDataBag.ACORN

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.SLA := DBData.QDataBag.SLA

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.LAC := DBData.QDataBag.LAC

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.MAFF := DBData.QDataBag.MAFF

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.V2 := DBData.QDataBag.V2

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.Padding := DBData.QDataBag.Padding

COMPUTE IF: *QDataBag.Serial = EMPTY*
AND: *DBData.SEARCH (Serial)*

QDataBag.Serial := DBData.QDataBag.Serial

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

CheckYear := 1999

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

CheckYear := 1998

COMPUTE IF: *SCPR = Yes*
AND: *FRAC (QSerial.Area / 2) > 0*

QDataBag.V2 := 1

COMPUTE IF: *SCPR = Yes*
AND: *NOT (FRAC (QSerial.Area / 2) > 0)*

QDataBag.V2 := 2

COMPUTE IF: *NOT (SCPR = Yes)*
AND: *FRAC (QSerial.Area / 2) > 0*

QDataBag.V2 := 3

COMPUTE IF: *NOT (SCPR = Yes)*
AND: *NOT (FRAC (QSerial.Area / 2) > 0)*

QDataBag.V2 := 4

RECORD ALWAYS:

Scotland

QUESTION ONLY VISIBLE FOR TESTING PURPOSES:

Is this a Scottish sample point?

(1) Yes
(2) No

COMPUTE IF: *QDataBag.SSTRTReg IN [22 .. 26]*

Scotland := Yes

COMPUTE IF: *NOT (QDataBag.SSTRTReg IN [22 .. 26])*

Scotland := No

ASK ALWAYS:

First

INTERVIEWER: FOR INFORMATION... You are in the Household Schedule for
Address No: ^QSerial.Address
Household No: ^QSerial.Hhold
- TO GO DIRECTLY TO 'ADMIN', PRESS <Ctrl + Enter>.
- TO CONTINUE WITH INTERVIEW PRESS '1' AND <Enter>.

Questionnaire Version: ^IVers

(1) Continue

ASK IF: *SCPR = Yes*

Address

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

RECORD ALWAYS:

MaxRef

Highest occurrence of REFUSAL for a specified set of questions for all the times the questionnaire has been open.

0..97

RECORD ALWAYS:

CurrRef

Current occurrence of REFUSAL for a specified set of questions.

0..97

FRS35.QSignIn

ASK ALWAYS:

StartDat

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

DATE

RECORD ALWAYS:

IntSTime

Interview start time

TIME

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

IntSTime := STARTTIME

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

Editor

EDITOR at HQ: Enter your identification number.

1..97

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

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.

RECORD ALWAYS:

DLYear

DATE

COMPUTE ALWAYS:

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

RECORD ALWAYS:

hhchu11

(1) Yes
(2) No

RECORD ALWAYS:

HHSize

Household size including any x-ed out

0..14

FRS35.QNames

Names of household members

ASK ALWAYS:

WhoHere

Who normally lives at this address?

(1) Press <Enter> to continue.

FRS35.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

FRS35.QNames (continued)

Names of household members

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

DMName[[Pers] := M[Pers].Name

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

HSize := Pers

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

WARN ALWAYS:

RESERVECHECK

RESERVECHECK

COMPUTE ALWAYS:

```
HHSize := QNames.HSize
```

FRS35.HHG

Data on adults in household

COMPUTE IF: *HHSize > 0*

FHHSize := PHHSize

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]

FRS35.HHG.P[]

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

BenUnit

HHG
Benefit Unit number.

0..14

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

Person

HHG
Person number in Household Grid.

0..14

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

Sex

HHG

INTERVIEWER: CODE SEX.

(1) Male
(2) Female

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

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) MEN ARE AGED 16-64 OR 65+
B) 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

Is ^LName ...READ OUT...
CODE FIRST TO APPLY...
HELP <F9>

(1) ...single, that is, never married,
(2) ...married and living with husband/wife, 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: *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*
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
     AND: Sex = Female
```

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'

FRS35.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

FRS35.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
     AND: QRel[R1].R IN [Spouse .. Cohabit]
     AgeOf > 15
```

Are you sure?
^LName is only ^AgeOf years old. Please check.

```
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: QRel[R1].R = Spouse
     MS = Marr
```

You've recorded ^LName as the spouse of ^QNames.M[R1].Name, but not as being 'Married & living with spouse'. Please amend one or the other

---

```
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
     MS <> Marr
```

You've recorded ^LName as 'cohabiting' with ^QNames.M[R1].Name, but also as 'MARRIED & living with spouse'. Please amend one or the other

---

```
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
     AgeOf > 15
```

You've coded ^LName as a parent (inc. foster/in-law/step) or grandparent, but ^heshe is less than 16 years old. Please check ^LName's age.

```
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
```

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 mentioned at Hhlder?

(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 .. 18]
```

FtEd

HHG

Is ^LName presently in full-time education?

HELP <F9>

(1) Yes
(2) No

```
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

At what age did ^LName complete continuous full-time education^StillEduc?

^CONTINUOUS

GIVE ESTIMATE IF AGE NOT KNOWN.

ENTER '97', IF NEVER RECEIVED FULL-TIME EDUCATION.
HELP <F9>

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])*
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])*
TEA <> 96

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

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*
TEA<>NONRESPONSE

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).

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: *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*
((TEA > 10) OR (TEA = 97)) OR TEA=NONRESPONSE

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*
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

What type of school or college does ^HeShe attend?

HELP <F9>

(2) Nursery/primary/playschool (state run)
(3) Special school state run (e.g. for the handicapped)
(4) Secondary school (state run or assisted)
(5) Non-advanced further education/6th form/tertiary/further education college
(6) Any PRIVATE school (nursery, prep, secondary)
(7) University/polytechnic/any other higher education

WARN IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *(TEA = 96) OR (FtEd = Yes)*
IN(AgeOf,[0])

Are you sure this person is in HIGHER education? This means they will form a separate Benefit Unit. Check whether anyone is still receiving Child Benefit for this person: if so, amend TypeEd to a lower code.

WARN IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *(TEA = 96) OR (FtEd = Yes)*
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)*
IN(AgeOf,[0])

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

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

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

Depend := Adult

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

Depend := Adult

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

Depend := DepAd

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 15]) AND (AgeOf = RESPONSE)
```

Depend := Child

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT ((AgeOf IN [0 .. 15]) AND (AgeOf = RESPONSE)
```

Depend := Adult

```
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

FRS35.HHG (continued)

Data on adults in household

COMPUTE IF: *HHSize > 0*

P[1].LiveWith := No

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].NumPart := 0

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].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*

P[P1].LiveWith := Yes

```
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
```

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
     (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
     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[P1].QRel[P2].R = Spouse
     ((P[P1].MS = Marr) OR P[P1].MS=EMPTY) AND
     INVOLVING(P[P2].QRel[P1].R,P[P1].MS)
```

You've recorded ^P[P1].Name as the spouse of ^P[P2].Name, but not 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
     (P[P1].MS <> Marr) AND INVOLVING(P[P2].QRel[P1].R,P[P1].MS)
```

You've recorded ^P[P1].Name as 'cohabiting' with ^P[P2].Name, but also 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*
(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*
(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*
(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.

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)

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])) 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*
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*
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 IN [DepAd .. Child]) 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 IN [DepAd .. Child]) 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.

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***WARN IF**: HHSize > 0*
RESERVECHECK

RESERVECHECK

***ASK 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)*

SpOut

INTERVIEWER:
You've recorded ^DMName[Loop1] as 'Married & living with spouse', but without a spouse in the household. PLEASE CHECK THIS. If spouse is away for six months or more, press 1 and enter to continue.
BUT IF NOT (eg if separated), ENTER 2 AND AMEND HOUSEHOLD GRID.

(1) Married, spouse not in household
(2) Other - AMEND 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)*
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

***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 := ''

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

AllAd

Number of adults 16+ less x-ed out

0..14

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

AllCh

Number of children < 16 less x-ed out

0..14

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

AllPers

Number of household members less x-ed out

0..14

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

AllElig

Number of adults eligible for BU interview

0..14

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

ChRegis

Children registered creche, childminder, playgroup

0..14

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*

AllElig := 0

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

ChRegis := 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]*

OneHoH := 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].Depend = Adult*

AllElig := (AllElig + 1)

FRS35.PRec[]

Person record from HH Grid as AUXFIELDS for global use.

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

Sex

(1) Male
(2) Female

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

MS

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

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

FtEd

(1) Yes
(2) No

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

TypeEd

(2) Nursery/primary/playschool (state run)
(3) Special school state run (e.g. for the handicapped)
(4) Secondary school (state run or assisted)
(5) Non-advanced further education/6th form/tertiary/further education college
(6) Any PRIVATE school (nursery, prep, secondary)
(7) University/polytechnic/any other higher education

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

Depend

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

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

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].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]) AND (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 (LegGuard[Loop1] = 2)*

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

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

AllPers := (AllAd + AllCh)

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'

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*

RentName := ''

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

HHG.P[Loop1].Hholder := Yes

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

HHG.P[Loop1].Hholder := No

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

RESERVECHECK

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

RESERVECHECK

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

RESERVECHECK

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

RESERVECHECK

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

RESERVECHECK

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

HoHPrtnr

Person number of HoH's spouse/partner.

1..15

RECORD IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(AllAd = 1) AND (OneHoH = RESPONSE)*

HoHNum

INTERVIEWER: ENTER PERSON NUMBER OF HEAD OF HOUSEHOLD, USING STANDARD RULES (SEE INSTRUCTIONS OR HELP <F9>).
^AdNameNo

0..14

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(AllAd = 1) AND (OneHoH = RESPONSE)*

HoHNum := OneHoH

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(AllAd = 1) AND (OneHoH = RESPONSE)*

HoHPrtnr := 15

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *NOT ((AllAd = 1) AND (OneHoH = RESPONSE)*

HoHNum

INTERVIEWER: ENTER PERSON NUMBER OF HEAD OF HOUSEHOLD, USING STANDARD RULES (SEE INSTRUCTIONS OR HELP <F9>).
^AdNameNo

0..14

CHECK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *NOT ((AllAd = 1) AND (OneHoH = RESPONSE)*
PRec[HoHNum].Depend = Adult

Code ^HoHNum is not valid for this question.

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *NOT ((AllAd = 1) AND (OneHoH = RESPONSE)*
NOT((PRec[HoHNum].Sex = Female) AND (PRec[HoHNum].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.)

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*

HoHNames := DMName[HoHNum]

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

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

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

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

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

HoHPrtnr := Loop1

FRS35.QEthnic

Ethnic data on adults in household

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*

P[Personnr].PersId := Personnr

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*

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

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Personnr := 1 TO HHSize*
AND: *PRec[Personnr].Depend = Adult*

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

FRS35.QEthnic.P[]

```
RECORD IF: HoHNum = RESPONSE
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

BenUnit

QEthnic

Benefit Unit number.

1..7

```
RECORD IF: HoHNum = RESPONSE
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

PersId

QEthnic

Person identifier.

0..14

```
DISPLAY IF: HoHNum = RESPONSE
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

EName

QEthnic

STRING[15]

```
ASK IF: HoHNum = RESPONSE
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
```

EthGrp

QEthnic

SHOW CARD A
To which of these groups do you consider ^EName belongs?

INTERVIEWER: THIS IS A QUESTION OF OPINION.

(1) White
(2) Black - Caribbean
(3) Black - African
(4) Black - Other Black Groups
(5) Indian
(6) Pakistani
(7) Bangladeshi
(8) Chinese
(9) None of these

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

WARN IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

WARN IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

WARN IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

WARN IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

WARN IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

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

ABen[Loop1] := 1

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

RECORD IF: *HoHNum = RESPONSE*

NewBU

Amended number of B.U:s (only confirmed h'hold members)

0..7

COMPUTE IF: *HoHNum = RESPONSE*

ABen[HoHNum] := 1

COMPUTE IF: *HoHNum = RESPONSE*

Last := 1

COMPUTE IF: *HoHNum = RESPONSE*
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: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)*

Last := (Last + 1)

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)*

ABen[Loop1] := Last

COMPUTE IF: *HoHNum = RESPONSE*
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: *HoHNum = RESPONSE*
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: *HoHNum = RESPONSE*
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: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *NOT (DMParent2[Loop1] <> 0)*

Last := (Last + 1)

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *NOT (DMParent2[Loop1] <> 0)*

ABen[Loop1] := Last

COMPUTE IF: *HoHNum = RESPONSE*

NewBU := Last

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

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

NameInBU[Loop1] := ''

COMPUTE IF: *HoHNum = RESPONSE*
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: *HoHNum = RESPONSE*
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: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO NewBU*

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

ASK IF: *HoHNum = RESPONSE*

ShowBen

INTERVIEWER - THAT COMPLETES THE PERSONAL INFORMATION ABOUT THE INDIVIDUALS IN THIS HOUSEHOLD.
THE HOUSEHOLD MEMBERS HAVE BEEN ALLOCATED TO BENEFIT UNITS AS FOLLOWS:

B.U. MEMBERS ^NameInBU[1]^NameInBU[2]^NameInBU[3]^NameInBU[4]^NameInBU[5]
^NameInBU[6]^NameInBU[7]
TOTAL NUMBER OF BENEFIT UNITS = ^NewBU HELP <F9>

(1) PRESS <1> AND <ENTER> TO CONTINUE

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HoHNum = RESPONSE*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *HoHNum = RESPONSE*

hhchu11 := No

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Sex = RESPONSE*

DMBU[Loop1] := ABen[Loop1]

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Sex = RESPONSE*

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

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(DMBU[Loop1] = 1) AND (DMAge[Loop1] IN [0 .. 10])*

hhchu11 := Yes

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *Loop1 IN HHldr*

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

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *Loop1 IN HHldr*
AND: *DMBU[Loop1] <> 1*

NotHoHBU := 1

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

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

COMPUTE IF: *HoHNum = RESPONSE*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(Loop1 IN HHldr) OR (PRec[Loop1].Depend = Adult)*
AND: *NOT (BUAdName[DMBU[Loop1]] =)*

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

FRS35.QAccomdat

Questions about accommodation

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

Tenure

QAccomDat

SHOW CARD B
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 (shared ownership)
(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*
Tenure = RESPONSE

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.

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

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

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

SOBuy

QAccomDat

INTERVIEWER, ASK OR CODE:

SHARED OWNERS: 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

How many rooms do you have altogether in your accommodation, that's excluding bathrooms and toilets, but including kitchens?

INTERVIEWER: 'YOUR ACCOMMODATION' MEANS THE ACCOMMODATION OCCUPIED BY THIS HOUSEHOLD. EXCLUDE ANY ROOMS LET/SUBLET TO OTHER HOUSEHOLDS. CONSULT INSTRUCTIONS FOR TREATMENT OF EQUIVOCAL ROOMS, eg. attics, conservatories, basements.

HELP <F9>

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

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

BusRoom

QAccomDat

Are any of the rooms you have mentioned used for business because you are self-employed?

'YOU' = HoH/HOUSEHOLDER, OR SPOUSE/PARTNER

(1) Yes
(2) No

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
BusRoom <> Yes

THERE ARE ROOMS USED EITHER WHOLLY OR PARTLY FOR BUSINESS.
PLEASE FOLLOW THE EDIT INSTRUCTIONS.

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

OnBsRoom

QAccomDat

How many rooms are used ...READ OUT...

i) wholly for business?

0..10

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

PtBsRoom

QAccomDat

How many rooms are used ...READ OUT...

ii) partly for business?

0..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

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

Number of rooms partly used for business ^ChkTxt

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

Number of rooms only used for business ^ChkTxt

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
(OnBsRoom + PtBsRoom) <= Rooms

Number of rooms only or partly used for business ^ChkTxt

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

MainAcc

QAccomDat

INTERVIEWER CODE: IS THE HOUSEHOLD'S ACCOMMODATION...

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

```
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 := 'terraced/end of terrace?'

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

```
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 := 'some other kind of accommodation?'

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

TypeAcc

QAccomDat

INTERVIEWER CODE: IS IT...

(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]*
(((IN(TypeAcc,[Detached..Terraced])) AND (MainAcc = HseBun)) OR ((IN(TypeAcc,[Purpose_built,Converted_house])) AND (MainAcc = FltMas))) OR ((MainAcc = Oth) AND (IN(TypeAcc,[Mobile_home..Other_kind])))

This code is not valid for this accommodation.

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *MainAcc IN [HseBun .. FltMas, Oth]*
(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 'Rooms' code 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 the main living part of the accommodation?

(1) Basement/semi-basement
(2) Ground floor/street level
(3) 1st floor
(4) 2nd floor
(5) 3rd floor
(6) 4th-9th floor
(7) 10th floor or higher

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

Entry

QAccomDat

INTERVIEWER CODE: IS THERE AN ENTRYPHONE?

(1) Yes
(2) No

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

YearLive

QAccomDat

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

ENTER NUMBER OF MONTHS, TO NEAREST WHOLE MONTH.

0..11

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

HHStat

QAccomDat

INTERVIEWER : CLASSIFY THIS HOUSEHOLD AS ONE OF THE FOLLOWING:

(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 HoH is unclear or arbitrary - eg. students, nurses, unrelated adults etc, sharing ON EQUAL BASIS

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:

(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 HoH 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

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

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

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

FRS35.QRenting

Questions about renters

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*

Landlord

QRenting

Who is your landlord?

HELP <F9>

(1) The local authority/council/New Town development/Scottish Homes
(2) A housing association or co-operative or charitable trust
(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

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
Landlord = RESPONSE

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 IF: *QAccomdat.Tenure IN [Part .. Squatting]*

Housing_Benefit := 'Housing Benefit/ rent rebate/ 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)
```

YStart

QRenting

In which year did you first become a tenant of this accommodation?

INTERVIEWER: 'YOU'=PERSON(S) NAMED AT 'HHolder', 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: YStart IN [ToFeb97 .. AftMar97]
```

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?

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: YStart IN [ToFeb97 .. AftMar97]
     AND: Ctract IN [Signed .. NotSign]
     AND: YStart = ToFeb97
```

Short1

QRenting

There is a form of tenancy called a shorthold. It is for a fixed period and you had to be given a notice in writing by the landlord that told you it was a shorthold tenancy agreement. Here is an example of a notice to a tenant saying that the agreement is an assured shorthold. SHOW EXAMPLE OF NOTICE.

Does your agreement or notice state that it is a 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: (ResLL = No) OR (ResLL2 = No)
     AND: YStart IN [ToFeb97 .. AftMar97]
     AND: Ctract IN [Signed .. NotSign]
     AND: YStart = AftMar97
```

Short2

QRenting

Most tenancies are assured shortholds which are for a fixed period. There are others, just called 'assured', which are not for a fixed period. For these you have to be given a notice in writing by the landlord that tells you it is NOT an assured shorthold tenancy agreement. SHOW EXAMPLE OF NOTICE.

Does your agreement or notice state that it is NOT an assured shorthold?

(1) Not an assured shorthold
(2) Other agreement

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: YStart = Bef1988
```

FairRent

QRenting

Most rents are agreed privately between landlord and tenant. Sometimes the tenant can apply to the local rent officer or rent assessment committee to decide on a fair rent which is then registered. Has your rent for this accomodation been registered as a fair rent in this way, or not?

(1) Yes
(2) No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: ((((((ResLL = Yes) AND ResLL2 = EMPTY) OR (ResLL2 = Yes)) OR
     (Short1 = Other)) OR Short1 = NONRESPONSE) OR (Short2 = Other)) OR
     Short2 = NONRESPONSE
```

OthWay

QRenting

There are various ways in which landlords can let accommodation.
Will you please look at this card and tell me if your letting is one of these? SHOW CARD C

CODE FIRST THAT APPLIES.

(1) Company licence
(2) College licence
(3) Non-exclusive occupancy agreement
(4) Holiday let
(5) Low season let
(6) None of these

```
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 ' + HoHNames + ',')

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

How much rent do^es_household currently pay?

HELP <F9>

0.00..999997.00

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*
Rent = RESPONSE

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.

```
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)

FRS35.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

FRS35.QRenting (continued)

Questions about renters

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

RentWkly

QRenting

Standardised weekly amount

0.01..999997.00

```
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 <= 100) AND INVOLVING(RentPd,Rent)
```

This comes to œ^RentWkly a week.
Rents for Council tenants are normally below œ100 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 < 122) 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
```

MissVar := (MissVar + 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, AND YOU CANNOT ESTABLISH A SEPARATE AMOUNT FOR THE DOMESTIC 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
```

MissVar := (MissVar + 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'

RECORD IF: *QAccomdat.Tenure IN [Part .. Squatting]*

HBFlag3

QRenting

(1) One
(2) Two
(3) Three
(4) Four
(5) Five
(6) Six
(7) Seven
(8) Eight
(9) Nine

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*

HBenefit

QRenting

Are you ^allowed Housing Benefit or rent rebate, to help with paying your rent^directly?

(1) Yes
(2) No

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
HBenefit = RESPONSE

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.

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]

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*

HBenAmt

QRenting

How much ^Housing_Benefit are you allowed?

0.01..997.00

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *HBenefit = Yes*
AND: *HBenAmt = NONRESPONSE*

MissVar := (MissVar + 1)

```
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)

FRS35.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

FRS35.QRenting (continued)

Questions about renters

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: HBenPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

HBenWkly

QRenting

Standardised weekly amount

0.01..997.00

```
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 < 105) 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.

```
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

```
CHECK 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)
```

The amount you recorded for Housing Benefit included in the rent is greater than the rent recorded. Please amend your answers.

```
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/ rent rebate?

0.00..999997.00

```
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
```

HBWeeks

QRenting

For how many weeks have you been on ^Housing_Benefit (this time)?

ENTER TO NEAREST WHOLE WEEK (IF 97 OR MORE, ENTER 97).

1..97

```
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

MissVar := (MissVar + 1)
```

```
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)

FRS35.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

FRS35.QRenting (continued)

Questions about renters

```
RECORD 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

QRenting

Standardised weekly amount

0.01..1000.00

```
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 < 109) 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 (Scotland <> Yes)) AND Rent <>
     EMPTY
```

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 (Scotland <> Yes)) AND Rent <>
     EMPTY
     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 (Scotland <> Yes)) AND Rent <>
     EMPTY
     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 (Scotland <> Yes)) AND Rent <>
     EMPTY
     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 (Scotland <> Yes)) AND Rent <>
     EMPTY
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (WSInc = Sewer)

water_sewerage := 'water/sewerage'
```

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

WSIncAmt

QRenting

How much was included for ^water_sewerage?^COMBINED_AMOUNT^Consult_the_document

1.00..100.00

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
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!

FRS35.QRenting.Weekly()

Procedure Call

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[1] := 1

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[2] := 2

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[3] := 3

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[4] := 4

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[5] := 4.333

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[7] := 8.67

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[8] := 6.5

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[9] := 5.78

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[10] := 5.2

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[13] := 13

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[26] := 26

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*

PdConW[52] := 52

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*
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 (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *WSInc IN [Both, Water, Sewer]*
AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS35.QRenting (continued)

Questions about renters

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: ((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <>
     EMPTY
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND (RentPd IN [OneWeek .. Year])
     AND: LWeekly1 >= 1
```

WSIWkly

QRenting

Standardised weekly amount.

1.00..100.00

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: ((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <>
     EMPTY
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND (RentPd IN [OneWeek .. Year])
     AND: LWeekly1 >= 1
```

WSIWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: ((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <>
     EMPTY
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND (RentPd IN [OneWeek .. Year])
     AND: LWeekly1 >= 1
     AND: Edit = No
     (WSIWkly < 23) 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 (Scotland <> Yes)) AND Rent <>
     EMPTY
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSIncAmt = NONRESPONSE
```

MissVar := (MissVar + 1)

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((PTenure IN [Rents, Part]) AND (Scotland <> Yes)) AND Rent <> EMPTY*
AND: *Rent <> 0*

SerInc

QRenting

SHOW CARD D
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 (Scotland <> Yes)) 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?

SET [5] OF
(1) DSS
(2) Employer
(3) Other organisation
(4) Friend or relative
(5) Other

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*
NOT(IN(DSS,AccPay))

Are you sure? DSS only ever pay arrears of rent. Double-check, that respondent is not thinking of Housing Benefit.
If genuine arrears, suppress this warning.

FRS35.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] := 'DSS'

```
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] := '

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

RentSeq

QRenting

Rent sequence number

1..5

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*
AND: *In loop FOR Index := 1 TO 5*
AND: *Index IN AccPay*

RentSeq := PSeq

RECORD IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*
AND: *In loop FOR Index := 1 TO 5*
AND: *Index IN AccPay*

AccPay

QRenting

Who is that?
1. DSS
2. Employer
3. Other organisation
4. Friend or relative
5. Other)

1..5

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*

MissVar := (MissVar + 1)

```
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)

FRS35.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

FRS35.QRenting.QAccPay[] (continued)

```
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
     AND: AccPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

AccWkly

QRenting

Standardised weekly amount.

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 > 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 < 349) 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 ^PRent for rent, that you mentioned earlier, BEFORE or AFTER deducting this payment?

(1) Before
(2) After

FRS35.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: Edit = Yes
     AND: Rent = NONRESPONSE OR RentPd = NONRESPONSE
     NOT(IN(Landlord,[Council])) 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,[Council]))
```

MISSING INFORMATION FOR Housing Benefit AMOUNT AND/OR PERIOD.
FOLLOW EDIT INSTRUCTIONS FOR `Housing Benefit'

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *QAccomdat.Tenure = RentFree*
IN(QRenting.Landlord,[OrgEmpl..OthIndiv])

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

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *In loop FOR Loop1 := 1 TO HHSize*

Oldest := (MAX(DMAge[Loop1],Oldest))

FRS35.QOwner1

Questions about mortgages

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*

BuyYear

QOwner1

In which year did you buy this accommodation?

HELP <F9>

1900..1999

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.

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *QDataBag.SampMnth IN [4 .. 12]*
BuyYear <> 1999

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?

100..99999997

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *Edit = No*
PurcAmt < 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: *(PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy = StillM))*
AND: *PurcAmt = NONRESPONSE*

MissVar := (MissVar + 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

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*
AND: *OthMort3 = Yes*

OthPur3

QOwner1

SHOW CARD E

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))

This should only apply to to loans for purchase. Please resolve, or make a Note.

FRS35.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
              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 := ('

         (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
               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 := ('

         (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

THE NEXT QUESTIONS ARE ABOUT THE ^MORTGAGE ^INSTRUC

(1) Press <Enter> to continue.

```
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[])))
```

MortSeq

QOwner1

Mortgage sequence number.

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[])))
```

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

```
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?

1900..1999

```
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 <> 1999
```

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
```

BorrAmt

QOwner1

What was the original amount of this mortgage or loan^fill?

HELP <F9>

100..99999997

```
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
     BorrAmt < 190000
```

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
     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
```

MissVar := (MissVar + 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))
```

MissVar := (MissVar + 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

Since ^PBuyYear, have you taken out a re-mortgage, with a different lender - or have you extended the original loan by taking out a further advance or top-up?

INTERVIEWER: IF UNSURE, SEE HELP SCREEN <F9>.

IF BOTH, CODE 'YES' AND AT NEXT QUESTIONS, CODE FOR WHICHEVER WAS MOST RECENT.

(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 re-mortgage/further advance?

IF RE-MORTGAGED MORE THAN ONCE, TAKE THE MOST RECENT OCCASION.

1900..1999

```
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 <> 1999
```

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
     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?

100..9999997

```
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 < 190000
```

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
```

```
MissVar := (MissVar + 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 E
Which of these items best describe the reasons why you took out a re-mortgage/ further advance? 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.)

```
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
```

Lender

QOwner1

Who currently provides this mortgage or loan^to_buy_this_house: is it
...READ OUT (RUNNING PROMPT)... HELP <F9>

(1) ...a building society,
(2) ...a bank,
(3) ...other? (DESCRIBE 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*

MortType

QOwner1

Is this mortgage/loan... ...READ OUT (RUNNING PROMPT)...

(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 or Unit Trust mortgage,
(5) ...or both an ENDOWMENT (or any interest only) mortgage AND a REPAYMENT mortgage?
HELP <F9>

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*
MortType = RESPONSE

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 IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR i := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPPurcLoan = Two) AND (i = 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*

MortEnd

QOwner1

In what year is this mortgage due to be paid off?

1998..2999

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 <= (PBuyYear + 40)

Are you sure? The end-date would not normally be more than 40 years after the start date.
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*
(MortEnd <= (RMortYr + 40)) OR (RMortYr <> RESPONSE)

Are you sure? The end-date would not normally be more than 40 years after the last re-mortgage.
Please check your figures.

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*
MortEnd >= PBuyYear

The first mortgage was taken out in ^PBuyYear, so it can't be due to be paid off before that. Please amend your answers.

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*
MortEnd >= RMortYr

A re-mortgage was taken out in ^RMortYr, so the mortgage can't be due to be paid off 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*

MortLeft

QOwner1

What is the amount still outstanding on your mortgage/loan from this source - that is, how much do you still have to pay off?

1..9999997

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*

MissVar := (MissVar + 1)

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]*
AND: *RMort = Yes*
ABS(MortLeft - RMAmt) <= 50

For this kind of mortgage, the amount outstanding should equal the total amount of the re-mortgage. Please check and amend, else explain 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: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP]
     AND: NOT (RMort = Yes)
     ABS(MortLeft - BorrAmt) <= 50
```

For this kind of mortgage, the amount outstanding should equal the amount originally borrowed. Please check and amend, else explain 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: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: NOT (MortType IN [Endow, Pension, PEP])
     AND: RMort = Yes
     MortLeft < RMAmt
```

For this kind of mortgage, the amount outstanding should be less than the amount of re-mortgage. Please check and amend, else explain 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: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: NOT (MortType IN [Endow, Pension, PEP])
     AND: NOT (RMort = Yes)
     MortLeft < BorrAmt
```

For this kind of mortgage, the amount outstanding should be less than the amount originally borrowed. Please check and amend, else 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: PBuyYear >= 1995
```

MortFlex

QOwner1

Some mortgages give the borrower the flexibility to make certain changes without asking their lender first, such as varying the amount or timing of payments, or taking breaks in some circumstances. Do you have a mortgage of this type?

HELP <F9>

(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: MortType IN [Endow, Pension .. EndRep]
```

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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay = RESPONSE
```

```
LastPay := STR(MorInPay)
```

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 .. EndRep]*
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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay = DONTKNOW*

MissVar := (MissVar + 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: *MortType IN [Endow, Pension .. EndRep]*
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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay = REFUSAL*

MissVar := (MissVar + 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 0
```

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)

FRS35.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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS35.QOwner1.QMortgage.M[] (continued)

```
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 .. EndRep]
     AND: MorInPay > 0
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

MorIWkly

QOwner1

Standardised weekly amount.

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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 0
     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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 0
     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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 0*
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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 0*
AND: *MorInPd IN [OneWeek .. Year]*
AND: *LWeekly > 0*
AND: *MortLeft = RESPONSE*
AND: *EPIntC <= 4*

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: *MortType IN [Endow, Pension .. EndRep]*
AND: *MorInPay > 0*
AND: *MorInPd IN [OneWeek .. Year]*
AND: *LWeekly > 0*
AND: *MortLeft = RESPONSE*
AND: *EPIntC >= 16*

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: MortType IN [Endow, Pension .. EndRep]
     AND: MorInPay > 0
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     ((EPIntC > 4) AND (EPIntC < 16)) AND INVOLVING(MorInPd,MorInPay)
```

The interest payments work roughly out 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.

```
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 .. EndRep]
     AND: MortType IN [Endow, EndRep]
```

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 .. EndRep]
     AND: MortType IN [Endow, EndRep]
     AND: Edit = No
     Menpol = Yes
```

Normally there would be an endowment policy, with an ENDOWMENT mortgage: please check.

```
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 .. EndRep]
     AND: (MortType IN [Pension, PEP]) OR (Menpol = No)
```

EndwPrin

QOwner1

ASK OR CODE: How is repayment of the original loan covered?

CODE ONE ONLY.

(1) payments into a Pension Plan (pension mortgage)
(2) payments into a PEP (Personal Equity Plan)
(3) payments into a Unit Trust or Investment Trust scheme
(4) payments into any other savings/investment scheme
(5) None of the above (describe in a Note)

```
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 .. EndRep]
     AND: (MortType IN [Pension, PEP]) OR (Menpol = No)
     AND: EndwPrin = Pension
```

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: MortType IN [Endow, Pension .. EndRep]
```

MenPolAm0 := Yes

FRS35.QOwner1.QMortgage.M[].QEndow[]

```
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 .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

MortSeq

QOwner1

Mortgage sequence number.

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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

MortSeq := PPSeq

```
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 .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

EndowSeq

QOwner1

Endowment policy sequence number.

1..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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: *MortType IN [Endow, Pension .. EndRep]*
AND: *(Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])*
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: *MortType IN [Endow, Pension .. EndRep]*
AND: *(Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *PMenpol = Yes*

payment_contribution := ('premium on the' + next + ' 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 IN [Endow, Pension .. EndRep]*
AND: *(Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])*
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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: NOT (PMenpol = Yes)
```

payment_contribution := 'contribution to the (pension plan/PEP/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 IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

MenPolAm

QOwner1

How much was your last ^payment_contribution?

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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm = NONRESPONSE
```

MissVar := (MissVar + 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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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)

FRS35.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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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

FRS35.QOwner1.QMortgage.M[].QEndow[] (continued)

```
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 .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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

QOwner1

Standardised weekly amount

0.00..999.97

```
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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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 < 46) 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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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 (œ^PLastPay)?

(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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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?

1900..1999

```
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 .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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 <> 1999
```

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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PCount < 4
```

MpMore

QOwner1

Are there any more ^policies_plans covering the repayment of the mortgage or loan?

(1) Yes
(2) No

FRS35.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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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,QEndow[Count].MenPolPd)
```

The payment/endowment premium is included in the interest payment of œ^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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: QEndow[Count].MenPolAm <> 0
```

MenPolAm0 := 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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
```

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: MortType IN [Endow, Pension .. EndRep]
     AND: (Menpol = Yes) OR (EndwPrin IN [Pension, PEP, UnitT, Other])
     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: NOT (MortType IN [Endow, Pension .. EndRep])
```

IntPrPay

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 (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay = RESPONSE*

LastPay := STR(IntPrPay)

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 [Endow, Pension .. EndRep])*
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: *NOT (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay = DONTKNOW*

MissVar := (MissVar + 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 (MortType IN [Endow, Pension .. EndRep])*
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: *NOT (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay = REFUSAL*

MissVar := (MissVar + 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: NOT (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 0
```

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)

FRS35.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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])*
AND: *IntPrPay > 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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS35.QOwner1.QMortgage.M[] (continued)

```
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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 0
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IntPWkly

QOwner1

Standardised weekly amount.

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 (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 0
     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: NOT (MortType IN [Endow, Pension .. EndRep])
     AND: IntPrPay > 0
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: Edit = No
     (IntPWkly < 650) AND INVOLVING(IntPrPd,IntPrPay)
```

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: (MorInPay = EMPTY OR (MorInPay > 0)) AND (IntPrPay = EMPTY OR
     (IntPrPay > 0))
```

TaxRelf

QOwner1

Has standard rate tax relief on this mortgage/loan already been deducted from the payment you just mentioned (œ^LastPay)?

INTERVIEWER: INCLUDE ALL MIRAS ARRANGEMENTS

(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: 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

^Apart_do you have a mortgage protection policy, that would pay this mortgage/loan in the event of sickness, accident^redundancy?

HELP <F9>

(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]*
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*
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 (œ' + 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: EndwPrin IN [Pension, PEP, UnitT, Other]
```

PCP := (PCP + ' or in the (pension/PEP/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 (EndwPrin IN [Pension, PEP, UnitT, Other])
```

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: EndwPrin IN [Pension, PEP, UnitT, Other]
```

PCP := 'the (pension/PEP/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 (EndwPrin IN [Pension, PEP, UnitT, Other])
```

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: EndwPrin IN [Pension, PEP, UnitT, Other]
```

PC := 'pension/PEP/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 (EndwPrin IN [Pension, PEP, UnitT, Other])
```

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'

FRS35.QOwner1.QMortgage.M[].QMortProt[]

```
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 ***

How much was your last payment?

HELP <F9>

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
```

MissVar := (MissVar + 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: 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)

FRS35.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

FRS35.QOwner1.QMortgage.M[].QMortProt[] (continued)

```
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
     AND: IncMPPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IncMWkly

QOwner1

*** ^Order[Count] MORTGAGE PROTECTION POLICY ***

Standardised weekly amount.

0.00..997.99

```
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 > 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?

1900..1999

```
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 <> 1999
```

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 ^payment_contribution_premium?

(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 ^premium_contribution?

(1) mortgage payment
(2) ^premium_contribution

FRS35.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 (œ^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)
     QMortProt[Count].IncMPAmt<>NONRESPONSE
```

MISSING AMOUNT FOR Mortgage Protection Policy. NOTE THE SIZE OF LAST MORTGAGE PAYMENT, DISPLAYED BELOW, THEN FOLLOW Edit Instructions TO FILL IN IncMPAmt.

```
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) DSS
(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] := 'DSS'
```

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] := '

FRS35.QOwner1.QMortgage.M[].QOutside[]

```
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
```

MortSeq

QOwner1

Mortgage sequence number.

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: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

MortSeq := PPSeq

```
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
```

ContSeq

QOwner1

Mortgage contribution sequence.

1..6

```
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

```
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
```

OutsPay

QOwner1

Who is that?
1. DSS
2. Employer
3. Other organisation
4. Friend or relative
5. Mortgage protection/insurance policy
6. Other

1..6

```
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
```

MissVar := (MissVar + 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: 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)

FRS35.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

FRS35.QOwner1.QMortgage.M[].QOutside[] (continued)

```
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
     AND: OutsPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

OutWkly

QOwner1

Standardised weekly amount.

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 > 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

FRS35.QOwner1.QMortgage.M[] (continued)

```
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' = HoH/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) Local Authority or Council (incl. GLC)
(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

```
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 E

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))
```

This should only apply to to loans for purchase. 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[])))
     EndwPrin <> None
```

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[])))*
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[])))*
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[])))*
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)*
RMAmt >= BorrAmt

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[])))*
AND: *(MortType = Pension) AND (EndwPrin = RESPONSE)*
EndwPrin = Pension

This method of capital repayment (at EndwPrin) 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: *(MortType = PEP) AND (EndwPrin = RESPONSE)*
IN(EndwPrin,[PEP,UnitT,Other])

This method of capital repayment (at EndwPrin) 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[])))*
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

FRS35.QOwner1.QMortgage

```
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[])))
     M[i].BorrAmt <= PPurcAmt
```

The amount borrowed is more than the purchase price - this is very unusual. Please check your figures and, if necessary, explain in a Note.

FRS35 (continued)

FAMILY RESOURCES SURVEY 1998/99

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

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
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*
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*
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*
AND: *(QAccomdat.Tenure = Part) AND (QAccomdat.SOBuy = Paid)*

AskStruc := 2

FRS35.QInsur

Questions about structure insurance.

ASK IF: *QAccomdat.HHStat <> EMPTY*
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*
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?

FRS35.QInsur.QStructure[]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: (PStrCov = Struct) OR (PCovOths = Struct)

combined := ' structure'
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: PStrCov = Furn

combined := ' contents'
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: (PStrCov = Combine) OR (PCovOths = Combine)

combined := ' combined'
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: PSeq = 1

included := ' included in your last mortgage payment'
```

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: NOT (PSeq = 1)

last := ' last'
```

```
ASK IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
```

StrAmt

QInsur

How much was the^last premium^included for this^combined policy?

0.01..9997.00

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt = NONRESPONSE
```

MissVar := (MissVar + 1)

```
ASK IF: QAccomdat.HHStat <> EMPTY
     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)

FRS35.QInsur.QStructure[].Weekly()

Procedure Call

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[1] := 1

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[2] := 2

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[3] := 3

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[4] := 4

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[5] := 4.333

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[7] := 8.67

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[8] := 6.5

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[9] := 5.78

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[10] := 5.2

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[13] := 13

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[26] := 26

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

PdConW[52] := 52

COMPUTE IF: *QAccomdat.HHStat <> EMPTY*
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*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*
AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS35.QInsur.QStructure[] (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY
     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
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (StrWkly < 26) 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.