

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

<i>FRS35</i> _____	23
<i>FRS35.QSerial</i> _____	24
<i>FRS35 (continued)</i> _____	26
<i>FRS35.QDataBag</i> _____	27
<i>FRS35 (continued)</i> _____	30
<i>FRS35.QSignIn</i> _____	34
<i>FRS35 (continued)</i> _____	35
<i>FRS35.QNames</i> _____	37
<i>FRS35.QNames.M[]</i> _____	38
<i>FRS35.QNames (continued)</i> _____	39
<i>FRS35 (continued)</i> _____	40
<i>FRS35.HHG</i> _____	41
<i>FRS35.HHG.P[]</i> _____	42
<i>FRS35.HHG.P[].QRel[]</i> _____	46
<i>FRS35.HHG.P[] (continued)</i> _____	48
<i>FRS35.HHG (continued)</i> _____	56
<i>FRS35 (continued)</i> _____	62
<i>FRS35.PRec[]</i> _____	66
<i>FRS35 (continued)</i> _____	68
<i>FRS35.QEthnic</i> _____	73
<i>FRS35.QEthnic.P[]</i> _____	74
<i>FRS35 (continued)</i> _____	76
<i>FRS35.QAccomdat</i> _____	83
<i>FRS35 (continued)</i> _____	94
<i>FRS35.QRenting</i> _____	95
<i>FRS35.QRenting.Weekly()</i> _____	102
<i>FRS35.QRenting (continued)</i> _____	104
<i>FRS35.QRenting.Weekly()</i> _____	109
<i>FRS35.QRenting (continued)</i> _____	111

<i>FRS35.QRenting.Weekly()</i> _____	115
<i>FRS35.QRenting (continued)</i> _____	117
<i>FRS35.QRenting.Weekly()</i> _____	120
<i>FRS35.QRenting (continued)</i> _____	122
<i>FRS35.QRenting.QAccPay[]</i> _____	125
<i>FRS35.QRenting.QAccPay[].Weekly()</i> _____	128
<i>FRS35.QRenting.QAccPay[] (continued)</i> _____	131
<i>FRS35.QRenting (continued)</i> _____	133
<i>FRS35 (continued)</i> _____	134
<i>FRS35.QOwner1</i> _____	135
<i>FRS35.QOwner1.QMortgage.M[]</i> _____	139
<i>FRS35.QOwner1.QMortgage.M[].Weekly()</i> _____	155
<i>FRS35.QOwner1.QMortgage.M[] (continued)</i> _____	159
<i>FRS35.QOwner1.QMortgage.M[].QEndow[]</i> _____	163
<i>FRS35.QOwner1.QMortgage.M[].QEndow[].Weekly()</i> _____	169
<i>FRS35.QOwner1.QMortgage.M[].QEndow[] (continued)</i> _____	173
<i>FRS35.QOwner1.QMortgage.M[] (continued)</i> _____	177
<i>FRS35.QOwner1.QMortgage.M[].Weekly()</i> _____	181
<i>FRS35.QOwner1.QMortgage.M[] (continued)</i> _____	185
<i>FRS35.QOwner1.QMortgage.M[].QMortProt[]</i> _____	192
<i>FRS35.QOwner1.QMortgage.M[].QMortProt[].Weekly()</i> _____	194
<i>FRS35.QOwner1.QMortgage.M[].QMortProt[] (continued)</i> _____	198
<i>FRS35.QOwner1.QMortgage.M[] (continued)</i> _____	201
<i>FRS35.QOwner1.QMortgage.M[].QOutside[]</i> _____	204
<i>FRS35.QOwner1.QMortgage.M[].QOutside[].Weekly()</i> _____	208
<i>FRS35.QOwner1.QMortgage.M[].QOutside[] (continued)</i> _____	212
<i>FRS35.QOwner1.QMortgage.M[] (continued)</i> _____	214
<i>FRS35.QOwner1.QMortgage</i> _____	220
<i>FRS35 (continued)</i> _____	221
<i>FRS35.QInsur</i> _____	223
<i>FRS35.QInsur.QStructure[]</i> _____	224
<i>FRS35.QInsur.QStructure[].Weekly()</i> _____	226
<i>FRS35.QInsur.QStructure[] (continued)</i> _____	228
<i>FRS35.QInsur (continued)</i> _____	229
<i>FRS35.QInsur.QStructure[]</i> _____	230
<i>FRS35.QInsur.QStructure[].Weekly()</i> _____	232

<i>FRS35.QInsur.QStructure[] (continued)</i>	234
<i>FRS35 (continued)</i>	235
<i>FRS35.QCounTax</i>	236
<i>FRS35.QCounTax.Weekly()</i>	245
<i>FRS35.QCounTax (continued)</i>	247
<i>FRS35 (continued)</i>	260
<i>FRS35.QAccomCharge</i>	262
<i>FRS35.QAccomCharge.QChargeAmtPd[]</i>	263
<i>FRS35.QAccomCharge (continued)</i>	265
<i>FRS35 (continued)</i>	266
<i>FRS35.QWaterSew</i>	268
<i>FRS35 (continued)</i>	273
<i>FRS35.QLodger</i>	274
<i>FRS35.QLodger.BordLodg[]</i>	276
<i>FRS35.QLodger.BordLodg[].Weekly()</i>	279
<i>FRS35.QLodger.BordLodg[] (continued)</i>	282
<i>FRS35.QLodger.BordLodg[].Weekly()</i>	285
<i>FRS35.QLodger.BordLodg[] (continued)</i>	288
<i>FRS35.QLodger.BordLodg[].ConIt</i>	289
<i>FRS35.QLodger (continued)</i>	290
<i>FRS35 (continued)</i>	291
<i>FRS35.QSharer</i>	292
<i>FRS35.QSharer.Sharer[]</i>	293
<i>FRS35.QSharer.Sharer[].Weekly()</i>	295
<i>FRS35.QSharer.Sharer[] (continued)</i>	298
<i>FRS35.QSharer.Sharer[].Weekly()</i>	300
<i>FRS35.QSharer.Sharer[] (continued)</i>	303
<i>FRS35.QSharer.Sharer[].ShareIt</i>	304
<i>FRS35.QSharer (continued)</i>	305
<i>FRS35 (continued)</i>	306
<i>FRS35.QProperty</i>	307
<i>FRS35 (continued)</i>	309
<i>FRS35.QPolicies</i>	310
<i>FRS35.QPolicies.Policy[]</i>	311
<i>FRS35 (continued)</i>	315
<i>FRS35.QModCons</i>	316

<i>FRS35 (continued)</i>	319
<i>FRS35.QTVehic</i>	320
<i>FRS35.QTVehic.QVehic[]</i>	322
<i>FRS35.QTVehic (continued)</i>	325
<i>FRS35 (continued)</i>	326
<i>FRS35.QWelfare</i>	329
<i>FRS35.QWelfare.QPres[]</i>	330
<i>FRS35.QWelfare (continued)</i>	342
<i>FRS35.QWelfare.QDent[]</i>	343
<i>FRS35.QWelfare (continued)</i>	355
<i>FRS35.QWelfare.QEye[]</i>	356
<i>FRS35.QWelfare (continued)</i>	368
<i>FRS35.QWelfare.QSpecs[]</i>	369
<i>FRS35.QWelfare (continued)</i>	381
<i>FRS35.QWelfare.QHosp[]</i>	382
<i>FRS35.QWelfare (continued)</i>	394
<i>FRS35.QWelfare.WMkQ[]</i>	396
<i>FRS35.QWelfare (continued)</i>	399
<i>FRS35.QWelfare.SMkQ[]</i>	400
<i>FRS35.QWelfare (continued)</i>	402
<i>FRS35.QWelfare.SMIQ[]</i>	403
<i>FRS35.QWelfare (continued)</i>	406
<i>FRS35 (continued)</i>	407
<i>FRS35.QChCare</i>	409
<i>FRS35.QChCare.Child[]</i>	411
<i>FRS35.QChCare (continued)</i>	422
<i>FRS35 (continued)</i>	423
<i>FRS35.QCare</i>	424
<i>FRS35.QCare.QRecHelp</i>	425
<i>FRS35.QCare.QRecHelp.Recip[]</i>	431
<i>FRS35.QCare (continued)</i>	443
<i>FRS35 (continued)</i>	444
<i>FRS35.Benefit_Unit[]</i>	449
<i>FRS35.Benefit_Unit[.QHealth1</i>	462
<i>FRS35.Benefit_Unit[.QHealth1.Adult[]</i>	463
<i>FRS35.Benefit_Unit[.QHealth1 (continued)</i>	466

<i>FRS35.Benefit_Unit[] (continued)</i> _____	467
<i>FRS35.Benefit_Unit[].QHealth2</i> _____	468
<i>FRS35.Benefit_Unit[].QHealth2.Child[]</i> _____	469
<i>FRS35.Benefit_Unit[] (continued)</i> _____	472
<i>FRS35.Benefit_Unit[].QHealth3</i> _____	473
<i>FRS35.Benefit_Unit[].QHealth3.Adult[]</i> _____	474
<i>FRS35.Benefit_Unit[].QHealth3 (continued)</i> _____	476
<i>FRS35.Benefit_Unit[] (continued)</i> _____	477
<i>FRS35.Benefit_Unit[].QEduci1</i> _____	478
<i>FRS35.Benefit_Unit[].QEduci1.Adult[]</i> _____	479
<i>FRS35.Benefit_Unit[].QEduci1 (continued)</i> _____	484
<i>FRS35.Benefit_Unit[] (continued)</i> _____	485
<i>FRS35.Benefit_Unit[].QEduci2</i> _____	486
<i>FRS35.Benefit_Unit[].QEduci2.Adult[]</i> _____	487
<i>FRS35.Benefit_Unit[].QEduci2.Adult[].QGrant[]</i> _____	488
<i>FRS35.Benefit_Unit[].QEduci2.Adult[].QGrant[]</i> _____	493
<i>FRS35.Benefit_Unit[].QEduci2.Adult[] (continued)</i> _____	499
<i>FRS35.Benefit_Unit[].QEduci2 (continued)</i> _____	501
<i>FRS35.Benefit_Unit[] (continued)</i> _____	502
<i>FRS35.Benefit_Unit[].QEduci3</i> _____	503
<i>FRS35.Benefit_Unit[].QEduci3.Adult[]</i> _____	504
<i>FRS35.Benefit_Unit[].QEduci3.Adult[].QEdLoan[]</i> _____	507
<i>FRS35.Benefit_Unit[].QEduci3.Adult[].QEdLoan[].Weekly()</i> _____	511
<i>FRS35.Benefit_Unit[].QEduci3.Adult[].QEdLoan[] (continued)</i> _____	515
<i>FRS35.Benefit_Unit[].QEduci3.Adult[].QEdLoan[]</i> _____	517
<i>FRS35.Benefit_Unit[].QEduci3.Adult[].QEdLoan[].Weekly()</i> _____	521
<i>FRS35.Benefit_Unit[].QEduci3.Adult[].QEdLoan[] (continued)</i> _____	525
<i>FRS35.Benefit_Unit[].QEduci3.Adult[] (continued)</i> _____	527
<i>FRS35.Benefit_Unit[].QEduci3 (continued)</i> _____	529
<i>FRS35.Benefit_Unit[] (continued)</i> _____	530
<i>FRS35.Benefit_Unit[].QEduci4</i> _____	531
<i>FRS35.Benefit_Unit[].QEduci4.Adult[]</i> _____	532
<i>FRS35.Benefit_Unit[].QEduci4 (continued)</i> _____	534
<i>FRS35.Benefit_Unit[] (continued)</i> _____	535
<i>FRS35.Benefit_Unit[].QChEduc</i> _____	536
<i>FRS35.Benefit_Unit[].QChEduc.Child[]</i> _____	537

<i>FRS35.Benefit_Unit[.QChEduc.Child[.QAdEduc</i>	538
<i>FRS35.Benefit_Unit[.QChEduc.Child[ (continued)</i>	544
<i>FRS35.Benefit_Unit[.QChEduc.Child[.QGrant[</i>	545
<i>FRS35.Benefit_Unit[.QChEduc.Child[.QGrant[</i>	551
<i>FRS35.Benefit_Unit[ (continued)</i>	558
<i>FRS35.Benefit_Unit[.QNHHCh</i>	560
<i>FRS35.Benefit_Unit[.QNHHCh.Child[</i>	561
<i>FRS35.Benefit_Unit[.QNHHCh.Child[.Weekly()</i>	565
<i>FRS35.Benefit_Unit[.QNHHCh.Child[ (continued)</i>	569
<i>FRS35.Benefit_Unit[.QNHHCh (continued)</i>	571
<i>FRS35.Benefit_Unit[ (continued)</i>	572
<i>FRS35.Benefit_Unit[.QCurst1.Adult[</i>	574
<i>FRS35.Benefit_Unit[.QCurst1</i>	587
<i>FRS35.Benefit_Unit[ (continued)</i>	588
<i>FRS35.Benefit_Unit[.QCurst2.Adult[</i>	589
<i>FRS35.Benefit_Unit[.QCurst2</i>	597
<i>FRS35.Benefit_Unit[ (continued)</i>	599
<i>FRS35.Benefit_Unit[.QPens1.Adult[</i>	600
<i>FRS35.Benefit_Unit[.QPens1</i>	602
<i>FRS35.Benefit_Unit[ (continued)</i>	603
<i>FRS35.Benefit_Unit[.QClaim</i>	604
<i>FRS35.Benefit_Unit[.QClaim.Adult[</i>	605
<i>FRS35.Benefit_Unit[.QClaim (continued)</i>	606
<i>FRS35.Benefit_Unit[ (continued)</i>	607
<i>FRS35.Benefit_Unit[.QJobDes[</i>	608
<i>FRS35.Benefit_Unit[.QJobDes[.Subj1</i>	611
<i>FRS35.Benefit_Unit[.QJobDes[.QMainJob</i>	618
<i>FRS35.Benefit_Unit[.QJobDes[ (continued)</i>	625
<i>FRS35.Benefit_Unit[.QJobDes[.Subj2</i>	628
<i>FRS35.Benefit_Unit[.QJobDes[ (continued)</i>	635
<i>FRS35.Benefit_Unit[.QJobDes[.Subj3</i>	636
<i>FRS35.Benefit_Unit[.QJobDes[ (continued)</i>	643
<i>FRS35.Benefit_Unit[ (continued)</i>	644
<i>FRS35.Benefit_Unit[.QEmpJob[</i>	647
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	651
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	654

<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	655
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	658
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	659
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	662
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	666
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	669
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	671
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	674
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	680
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	683
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	686
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	689
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	690
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	693
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	694
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	697
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	700
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	704
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	708
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	711
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	712
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	715
<i>FRS35.Benefit_Unit[.QEmpJob[.HoursF</i>	717
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	725
<i>FRS35.Benefit_Unit[ (continued)</i>	731
<i>FRS35.Benefit_Unit[.QEmpJob[</i>	733
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	737
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	740
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	741
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	744
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	745
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	748
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	752
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	755
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	757
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	760

<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	766
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	769
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	772
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	775
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	776
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	779
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	780
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	783
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	786
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	790
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	794
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	797
<i>FRS35.Benefit_Unit[.QEmpJob[.Weekly()</i>	798
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	801
<i>FRS35.Benefit_Unit[.QEmpJob[.HoursF</i>	803
<i>FRS35.Benefit_Unit[.QEmpJob[ (continued)</i>	811
<i>FRS35.Benefit_Unit[ (continued)</i>	817
<i>FRS35.Benefit_Unit[.QSelfJob[</i>	819
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[</i>	820
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[.Weekly()</i>	840
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[ (continued)</i>	844
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[.Weekly()</i>	848
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[ (continued)</i>	852
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[.SelfHr</i>	858
<i>FRS35.Benefit_Unit[.QSelfJob[.Adult[ (continued)</i>	866
<i>FRS35.Benefit_Unit[.QSelfJob[ (continued)</i>	868
<i>FRS35.Benefit_Unit[ (continued)</i>	869
<i>FRS35.Benefit_Unit[.QTravel1</i>	871
<i>FRS35.Benefit_Unit[.QTravel1.Adult[</i>	872
<i>FRS35.Benefit_Unit[.QTravel1 (continued)</i>	874
<i>FRS35.Benefit_Unit[ (continued)</i>	875
<i>FRS35.Benefit_Unit[.QTravel2</i>	876
<i>FRS35.Benefit_Unit[.QTravel2.Adult[</i>	877
<i>FRS35.Benefit_Unit[.QTravel2 (continued)</i>	885
<i>FRS35.Benefit_Unit[ (continued)</i>	886
<i>FRS35.Benefit_Unit[.QPens2</i>	888



<i>FRS35.Benefit_Unit[.QPens2.Adult[]</i> _____	889
<i>FRS35.Benefit_Unit[.QPens2 (continued)</i> _____	891
<i>FRS35.Benefit_Unit[] (continued)</i> _____	892
<i>FRS35.Benefit_Unit[.QPens3</i> _____	893
<i>FRS35.Benefit_Unit[.QPens3.Adult[]</i> _____	894
<i>FRS35.Benefit_Unit[.QPens3 (continued)</i> _____	897
<i>FRS35.Benefit_Unit[] (continued)</i> _____	898
<i>FRS35.Benefit_Unit[.QPens4.Adult[]</i> _____	899
<i>FRS35.Benefit_Unit[.QPens4</i> _____	900
<i>FRS35.Benefit_Unit[] (continued)</i> _____	901
<i>FRS35.Benefit_Unit[.QPens5</i> _____	902
<i>FRS35.Benefit_Unit[.QPens5.Adult[]</i> _____	903
<i>FRS35.Benefit_Unit[.QPens5 (continued)</i> _____	905
<i>FRS35.Benefit_Unit[] (continued)</i> _____	906
<i>FRS35.Benefit_Unit[.QPens6.Adult[]</i> _____	907
<i>FRS35.Benefit_Unit[.QPens6</i> _____	908
<i>FRS35.Benefit_Unit[] (continued)</i> _____	909
<i>FRS35.Benefit_Unit[.QPens7.Adult[]</i> _____	910
<i>FRS35.Benefit_Unit[.QPens7.Adult[.Weekly()</i> _____	914
<i>FRS35.Benefit_Unit[.QPens7.Adult[] (continued)</i> _____	917
<i>FRS35.Benefit_Unit[.QPens7</i> _____	919
<i>FRS35.Benefit_Unit[] (continued)</i> _____	920
<i>FRS35.Benefit_Unit[.QBenefit</i> _____	922
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQa</i> _____	923
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQa.Adult[]</i> _____	924
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQa (continued)</i> _____	930
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQb</i> _____	931
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQb.Adult[]</i> _____	932
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQb (continued)</i> _____	935
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQc</i> _____	936
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQc.Adult[]</i> _____	937
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQc (continued)</i> _____	939
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQd</i> _____	940
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQd.Adult[]</i> _____	941
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQd (continued)</i> _____	946
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQe</i> _____	947

<i>FRS35.Benefit_Unit[.QBenefit.QBenefQe.Adult]</i> _____	948
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQe (continued)</i> _____	951
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQf</i> _____	952
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQf.Adult]</i> _____	953
<i>FRS35.Benefit_Unit[.QBenefit.QBenefQf (continued)</i> _____	956
<i>FRS35.Benefit_Unit[.QBenefit (continued)</i> _____	957
<i>FRS35.Benefit_Unit[.QBenefit.Weekly()</i> _____	961
<i>FRS35.Benefit_Unit[.QBenefit (continued)</i> _____	964
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2]</i> _____	968
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ChBHow</i> _____	974
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ChBAmt</i> _____	979
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ChBAmt.Weekly()</i> _____	982
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ChBAmt (continued)</i> _____	985
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i> _____	987
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.GrdHow</i> _____	988
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.GrdAmt</i> _____	993
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.GrdAmt.Weekly()</i> _____	996
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.GrdAmt (continued)</i> _____	999
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i> _____	1001
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.InvHow</i> _____	1002
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.InvAmt</i> _____	1007
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.InvAmt.Weekly()</i> _____	1010
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.InvAmt (continued)</i> _____	1013
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i> _____	1015
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenHow</i> _____	1016
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i> _____	1021
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenAmt</i> _____	1022
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenAmt.Weekly()</i> _____	1025
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenAmt (continued)</i> _____	1028
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenUsual</i> _____	1030
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenUsual.Weekly()</i> _____	1034
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenUsual (continued)</i> _____	1037
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenComp</i> _____	1038
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenComp.QPenQ]</i> _____	1040
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.PenComp (continued)</i> _____	1048
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i> _____	1049

<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].PenCWho</i>	1050
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].PenMWho</i>	1052
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].PenAWho</i>	1054
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1056
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotHow</i>	1057
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotAmt</i>	1062
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotAmt.Weekly()</i>	1066
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotAmt (continued)</i>	1069
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotUsul</i>	1071
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotUsul.Weekly()</i>	1075
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotUsul (continued)</i>	1079
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1080
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotCmp</i>	1081
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotCmp.QPenQ[]</i>	1083
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMotCmp (continued)</i>	1092
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1093
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMCWho</i>	1094
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMMWho</i>	1096
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WMAWho</i>	1098
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1100
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenHow</i>	1101
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenAmt</i>	1106
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenAmt.Weekly()</i>	1110
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenAmt (continued)</i>	1113
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenUsul</i>	1115
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenUsul.Weekly()</i>	1119
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenUsul (continued)</i>	1123
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1124
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenCmp</i>	1125
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenCmp.QPenQ[]</i>	1127
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPenCmp (continued)</i>	1136
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1137
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPCWho</i>	1138
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPMWho</i>	1140
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WPAWho</i>	1142
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1144

<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WDisHow</i>	1145
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WDisAmt</i>	1150
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WDisAmt.Weekly()</i>	1154
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WDisAmt (continued)</i>	1157
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1159
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WWidHow</i>	1160
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WWidAmt</i>	1165
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WWidAmt.Weekly()</i>	1169
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].WWidAmt (continued)</i>	1172
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1174
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].SevHow</i>	1175
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].SevAmt</i>	1180
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].SevAmt.Weekly()</i>	1183
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].SevAmt (continued)</i>	1186
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1188
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DisWHow</i>	1189
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DisWAmt</i>	1194
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DisWAmt.Weekly()</i>	1197
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DisWAmt (continued)</i>	1200
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1202
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLACHow</i>	1203
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLACAmt</i>	1208
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLACAmt.Weekly()</i>	1212
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLACAmt (continued)</i>	1215
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLACWho</i>	1217
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1219
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLAMHow</i>	1220
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLAMAmt</i>	1225
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLAMAmt.Weekly()</i>	1229
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLAMAmt (continued)</i>	1232
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].DLAMWho</i>	1234
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1236
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].AtAlHow</i>	1237
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].AtAlAmt</i>	1242
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].AtAlAmt.Weekly()</i>	1246
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].AtAlAmt (continued)</i>	1249

<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.AtAlWho</i>	1251
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1253
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.CareWho</i>	1254
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1256
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.MobWho</i>	1257
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1259
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.AttWho</i>	1260
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1262
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAHow</i>	1263
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAAmt</i>	1268
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAAmt.Weekly()</i>	1271
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAAmt (continued)</i>	1274
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAUsual</i>	1276
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAUsual.Weekly()</i>	1280
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSAUsual (continued)</i>	1283
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1284
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSALoan</i>	1285
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSALoan.Weekly()</i>	1288
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.JSALoan (continued)</i>	1291
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1294
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISHow</i>	1295
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISAmt</i>	1300
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISAmt.Weekly()</i>	1303
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISAmt (continued)</i>	1306
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISUsual</i>	1308
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISUsual.Weekly()</i>	1312
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISUsual (continued)</i>	1315
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1316
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISLoan</i>	1317
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISLoan.Weekly()</i>	1320
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.ISLoan (continued)</i>	1323
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[ (continued)</i>	1326
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.FCHow</i>	1327
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.FCAmt</i>	1332
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.FCAmt.Weekly()</i>	1335
<i>FRS35.Benefit_Unit[.QBenefit.QBenef2[.FCAmt (continued)</i>	1338

<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1340
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IBHow</i>	1341
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IBAmt</i>	1346
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IBAmt.Weekly()</i>	1349
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IBAmt (continued)</i>	1352
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1354
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IDisHow</i>	1356
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IDisAmt</i>	1361
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IDisAmt.Weekly()</i>	1364
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].IDisAmt (continued)</i>	1367
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1369
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].MAllHow</i>	1370
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].MAllAmt</i>	1375
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].MAllAmt.Weekly()</i>	1378
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].MAllAmt (continued)</i>	1381
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1383
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].Fun6M</i>	1385
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1388
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].MGrntHow</i>	1389
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].MGrnt6M</i>	1394
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1397
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].OBenAmt</i>	1405
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].OBenAmt.Weekly()</i>	1408
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[].OBenAmt (continued)</i>	1411
<i>FRS35.Benefit_Unit[].QBenefit.QBenef2[] (continued)</i>	1413
<i>FRS35.Benefit_Unit[].QBenefit (continued)</i>	1426
<i>FRS35.Benefit_Unit[].QBenefit.QBenef4</i>	1429
<i>FRS35.Benefit_Unit[].QBenefit.QBenef4.Adult[]</i>	1430
<i>FRS35.Benefit_Unit[].QBenefit.QBenef4 (continued)</i>	1432
<i>FRS35.Benefit_Unit[].QBenefit (continued)</i>	1433
<i>FRS35.Benefit_Unit[].QBenefit.QBenef5[]</i>	1434
<i>FRS35.Benefit_Unit[].QBenefit.QBenef5[].UEPres</i>	1436
<i>FRS35.Benefit_Unit[].QBenefit.QBenef5[].UEAmt</i>	1438
<i>FRS35.Benefit_Unit[].QBenefit.QBenef5[].UEAmt.Weekly()</i>	1441
<i>FRS35.Benefit_Unit[].QBenefit.QBenef5[].UEAmt (continued)</i>	1444
<i>FRS35.Benefit_Unit[].QBenefit.QBenef5[] (continued)</i>	1446

<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.TUPres</i>	1447
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.TUAmt</i>	1449
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.TUAmt.Weekly()</i>	1452
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.TUAmt (continued)</i>	1455
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[] (continued)</i>	1457
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.FRPres</i>	1458
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.FRAmt</i>	1460
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.FRAmt.Weekly()</i>	1463
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.FRAmt (continued)</i>	1466
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[] (continued)</i>	1468
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.PSPres</i>	1469
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.PSAmt</i>	1471
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.PSAmt.Weekly()</i>	1474
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.PSAmt (continued)</i>	1477
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[] (continued)</i>	1479
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.ACCPres</i>	1480
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.ACCAmt</i>	1482
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.ACCAmt.Weekly()</i>	1485
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.ACCAmt (continued)</i>	1488
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[] (continued)</i>	1490
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.HSPres</i>	1491
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.HSAmt</i>	1493
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.HSAmt.Weekly()</i>	1496
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[.HSAmt (continued)</i>	1499
<i>FRS35.Benefit_Unit[.QBenefit.QBenef5[] (continued)</i>	1501
<i>FRS35.Benefit_Unit[.QBenefit (continued)</i>	1503
<i>FRS35.Benefit_Unit[.QBenefit.QComb[]</i>	1505
<i>FRS35.Benefit_Unit[] (continued)</i>	1506
<i>FRS35.Benefit_Unit[.QOInca4.Adult[]</i>	1507
<i>FRS35.Benefit_Unit[.QOInca4</i>	1509
<i>FRS35.Benefit_Unit[] (continued)</i>	1510
<i>FRS35.Benefit_Unit[.QOInca5.Adult[]</i>	1512
<i>FRS35.Benefit_Unit[.QOInca5</i>	1514
<i>FRS35.Benefit_Unit[] (continued)</i>	1515
<i>FRS35.Benefit_Unit[.QOInca6[]</i>	1516
<i>FRS35.Benefit_Unit[.QOInca6[.EmpPrev[]</i>	1518

<i>FRS35.Benefit_Unit[.QOIncA6[] (continued)</i>	1526
<i>FRS35.Benefit_Unit[.QOIncA6[.SpouPrev[]</i>	1527
<i>FRS35.Benefit_Unit[.QOIncA6[] (continued)</i>	1535
<i>FRS35.Benefit_Unit[.QOIncA6[.PersPen[]</i>	1536
<i>FRS35.Benefit_Unit[.QOIncA6[] (continued)</i>	1544
<i>FRS35.Benefit_Unit[.QOIncA6[.TuPen[]</i>	1545
<i>FRS35.Benefit_Unit[.QOIncA6[] (continued)</i>	1553
<i>FRS35.Benefit_Unit[.QOIncA6[.AnnPen[]</i>	1554
<i>FRS35.Benefit_Unit[.QOIncA6[] (continued)</i>	1562
<i>FRS35.Benefit_Unit[.QOIncA6[.TrustQ[]</i>	1563
<i>FRS35.Benefit_Unit[.QOIncA6[] (continued)</i>	1571
<i>FRS35.Benefit_Unit[] (continued)</i>	1573
<i>FRS35.Benefit_Unit[.QOIncA7.Adult[]</i>	1574
<i>FRS35.Benefit_Unit[.QOIncA7</i>	1578
<i>FRS35.Benefit_Unit[] (continued)</i>	1579
<i>FRS35.Benefit_Unit[.QOIncB3[]</i>	1581
<i>FRS35.Benefit_Unit[.QOIncB3[.MntSelf</i>	1585
<i>FRS35.Benefit_Unit[.QOIncB3[.MntSelf.Weekly()</i>	1587
<i>FRS35.Benefit_Unit[.QOIncB3[.MntSelf (continued)</i>	1590
<i>FRS35.Benefit_Unit[.QOIncB3[.MntSelf.Weekly()</i>	1593
<i>FRS35.Benefit_Unit[.QOIncB3[.MntSelf (continued)</i>	1597
<i>FRS35.Benefit_Unit[.QOIncB3[] (continued)</i>	1600
<i>FRS35.Benefit_Unit[.QOIncB3[.MntCSA</i>	1601
<i>FRS35.Benefit_Unit[.QOIncB3[.MntCSA.Weekly()</i>	1603
<i>FRS35.Benefit_Unit[.QOIncB3[.MntCSA (continued)</i>	1606
<i>FRS35.Benefit_Unit[.QOIncB3[.MntCSA.Weekly()</i>	1609
<i>FRS35.Benefit_Unit[.QOIncB3[.MntCSA (continued)</i>	1613
<i>FRS35.Benefit_Unit[.QOIncB3[] (continued)</i>	1616
<i>FRS35.Benefit_Unit[.QOIncB3[.QMaint.Maint[]</i>	1617
<i>FRS35.Benefit_Unit[.QOIncB3[.QMaint.Maint[.Weekly()</i>	1622
<i>FRS35.Benefit_Unit[.QOIncB3[.QMaint.Maint[] (continued)</i>	1626
<i>FRS35.Benefit_Unit[.QOIncB3[.QMaint.Maint[.Weekly()</i>	1629
<i>FRS35.Benefit_Unit[.QOIncB3[.QMaint.Maint[] (continued)</i>	1633
<i>FRS35.Benefit_Unit[.QOIncB3[.QMaint</i>	1636
<i>FRS35.Benefit_Unit[] (continued)</i>	1637
<i>FRS35.Benefit_Unit[.QOIncA8</i>	1638



<i>FRS35.Benefit_Unit[.QOIncA8.Adult]</i> _____	1639
<i>FRS35.Benefit_Unit[.QOIncA8.Adult].Weekly()</i> _____	1641
<i>FRS35.Benefit_Unit[.QOIncA8.Adult] (continued)</i> _____	1644
<i>FRS35.Benefit_Unit[.QOIncA8.Adult].Weekly()</i> _____	1647
<i>FRS35.Benefit_Unit[.QOIncA8.Adult] (continued)</i> _____	1650
<i>FRS35.Benefit_Unit[.QOIncA8] (continued)</i> _____	1651
<i>FRS35.Benefit_Unit[] (continued)</i> _____	1652
<i>FRS35.Benefit_Unit[.QOIncB1]</i> _____	1654
<i>FRS35.Benefit_Unit[.QOIncB1.Adult]</i> _____	1655
<i>FRS35.Benefit_Unit[.QOIncB1] (continued)</i> _____	1656
<i>FRS35.Benefit_Unit[] (continued)</i> _____	1657
<i>FRS35.Benefit_Unit[.QOIncB2.QAllow1]</i> _____	1658
<i>FRS35.Benefit_Unit[.QOIncB2]</i> _____	1660
<i>FRS35.Benefit_Unit[.QOIncB2.Weekly()]</i> _____	1661
<i>FRS35.Benefit_Unit[.QOIncB2] (continued)</i> _____	1664
<i>FRS35.Benefit_Unit[.QOIncB2.QAllow2]</i> _____	1665
<i>FRS35.Benefit_Unit[.QOIncB2] (continued)</i> _____	1667
<i>FRS35.Benefit_Unit[.QOIncB2.Weekly()]</i> _____	1668
<i>FRS35.Benefit_Unit[.QOIncB2] (continued)</i> _____	1671
<i>FRS35.Benefit_Unit[] (continued)</i> _____	1672
<i>FRS35.Benefit_Unit[.QOIncB6]</i> _____	1673
<i>FRS35.Benefit_Unit[.QOIncB6.Adult]</i> _____	1674
<i>FRS35.Benefit_Unit[.QOIncB6] (continued)</i> _____	1676
<i>FRS35.Benefit_Unit[] (continued)</i> _____	1677
<i>FRS35.Benefit_Unit[.QOIncB9]</i> _____	1678
<i>FRS35.Benefit_Unit[.QOIncB9.Adult]</i> _____	1679
<i>FRS35.Benefit_Unit[.QOIncB9.Adult].Weekly()</i> _____	1681
<i>FRS35.Benefit_Unit[.QOIncB9.Adult] (continued)</i> _____	1684
<i>FRS35.Benefit_Unit[.QOIncB9] (continued)</i> _____	1685
<i>FRS35.Benefit_Unit[] (continued)</i> _____	1686
<i>FRS35.Benefit_Unit[.QOIncB10]</i> _____	1688
<i>FRS35.Benefit_Unit[.QOIncB10.Adult]</i> _____	1689
<i>FRS35.Benefit_Unit[.QOIncB10] (continued)</i> _____	1691
<i>FRS35.Benefit_Unit[] (continued)</i> _____	1692
<i>FRS35.Benefit_Unit[.QOIncB11]</i> _____	1693
<i>FRS35.Benefit_Unit[.QOIncB11.Adult]</i> _____	1694

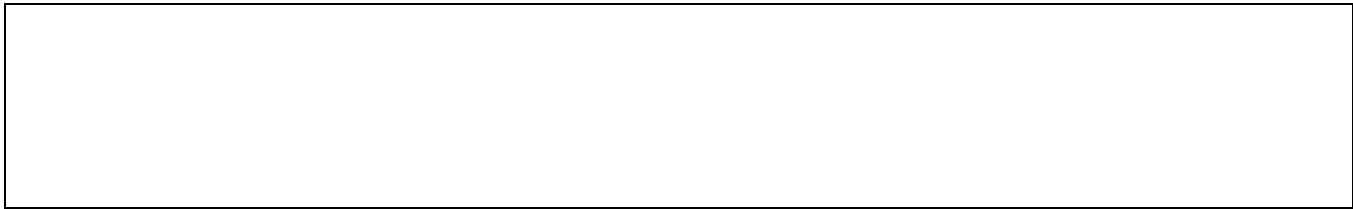
<i>FRS35.Benefit_Unit[.QOIncB11.Adult[.QBabMail</i>	1695
<i>FRS35.Benefit_Unit[.QOIncB11.Adult[ (continued)</i>	1698
<i>FRS35.Benefit_Unit[.QOIncB11 (continued)</i>	1699
<i>FRS35.Benefit_Unit[ (continued)</i>	1700
<i>FRS35.Benefit_Unit[.QOIncB12</i>	1701
<i>FRS35.Benefit_Unit[.QOIncB12.Adult[</i>	1702
<i>FRS35.Benefit_Unit[.QOIncB12.Adult[.QOddJob[</i>	1703
<i>FRS35.Benefit_Unit[.QOIncB12.Adult[ (continued)</i>	1707
<i>FRS35.Benefit_Unit[.QOIncB12 (continued)</i>	1708
<i>FRS35.Benefit_Unit[ (continued)</i>	1709
<i>FRS35.Benefit_Unit[.QChInc</i>	1711
<i>FRS35.Benefit_Unit[.QChInc.Child[</i>	1713
<i>FRS35.Benefit_Unit[.QChInc.Child[.Weekly()</i>	1716
<i>FRS35.Benefit_Unit[.QChInc.Child[ (continued)</i>	1719
<i>FRS35.Benefit_Unit[.QChInc.Child[.Weekly()</i>	1722
<i>FRS35.Benefit_Unit[.QChInc.Child[ (continued)</i>	1725
<i>FRS35.Benefit_Unit[.QChInc (continued)</i>	1726
<i>FRS35.Benefit_Unit[ (continued)</i>	1727
<i>FRS35.Benefit_Unit[.QAdInt1</i>	1729
<i>FRS35.Benefit_Unit[.QAdInt1.Adult[</i>	1730
<i>FRS35.Benefit_Unit[.QAdInt1 (continued)</i>	1731
<i>FRS35.Benefit_Unit[ (continued)</i>	1732
<i>FRS35.Benefit_Unit[.QAdult[</i>	1733
<i>FRS35.Benefit_Unit[ (continued)</i>	1734
<i>FRS35.Benefit_Unit[.QAdInt2</i>	1735
<i>FRS35.Benefit_Unit[.QAdInt2.Adult[.QIntrest[</i>	1736
<i>FRS35.Benefit_Unit[.QAdInt2.Adult[</i>	1742
<i>FRS35.Benefit_Unit[.QAdInt2 (continued)</i>	1743
<i>FRS35.Benefit_Unit[ (continued)</i>	1744
<i>FRS35.Benefit_Unit[.QAdInt3</i>	1746
<i>FRS35.Benefit_Unit[.QAdInt3.Adult[</i>	1747
<i>FRS35.Benefit_Unit[.QAdInt3 (continued)</i>	1748
<i>FRS35.Benefit_Unit[ (continued)</i>	1749
<i>FRS35.Benefit_Unit[.QAdInt4</i>	1751
<i>FRS35.Benefit_Unit[.QAdInt4.Adult[.QIntrest[</i>	1752
<i>FRS35.Benefit_Unit[.QAdInt4.Adult[</i>	1758

<i>FRS35.Benefit_Unit[.QAdInt4 (continued)</i>	1759
<i>FRS35.Benefit_Unit[] (continued)</i>	1760
<i>FRS35.Benefit_Unit[.QAdInt5</i>	1761
<i>FRS35.Benefit_Unit[.QAdInt5.Adult[]</i>	1762
<i>FRS35.Benefit_Unit[.QAdInt5 (continued)</i>	1763
<i>FRS35.Benefit_Unit[] (continued)</i>	1764
<i>FRS35.Benefit_Unit[.QChInt1</i>	1766
<i>FRS35.Benefit_Unit[.QChInt1.Child[]</i>	1767
<i>FRS35.Benefit_Unit[.QChInt1 (continued)</i>	1768
<i>FRS35.Benefit_Unit[] (continued)</i>	1769
<i>FRS35.Benefit_Unit[.QChild[]</i>	1770
<i>FRS35.Benefit_Unit[] (continued)</i>	1771
<i>FRS35.Benefit_Unit[.QChInt2.Child[]</i>	1772
<i>FRS35.Benefit_Unit[.QChInt2.Child[.QChIntr[]</i>	1773
<i>FRS35.Benefit_Unit[.QChInt2.Child[] (continued)</i>	1776
<i>FRS35.Benefit_Unit[.QChInt2.Child[.QChIntr[]</i>	1777
<i>FRS35.Benefit_Unit[.QChInt2.Child[] (continued)</i>	1780
<i>FRS35.Benefit_Unit[.QChInt2.Child[.QChIntr[]</i>	1781
<i>FRS35.Benefit_Unit[.QChInt2.Child[] (continued)</i>	1784
<i>FRS35.Benefit_Unit[.QChInt2.Child[.QChIntr[]</i>	1785
<i>FRS35.Benefit_Unit[.QChInt2.Child[] (continued)</i>	1788
<i>FRS35.Benefit_Unit[.QChInt2.Child[.QChIntr[]</i>	1789
<i>FRS35.Benefit_Unit[.QChInt2.Child[] (continued)</i>	1792
<i>FRS35.Benefit_Unit[.QChInt2.Child[.QChIntr[]</i>	1793
<i>FRS35.Benefit_Unit[] (continued)</i>	1796
<i>FRS35.Benefit_Unit[.QChInt3</i>	1797
<i>FRS35.Benefit_Unit[.QChInt3.Child[]</i>	1798
<i>FRS35.Benefit_Unit[.QChInt3 (continued)</i>	1799
<i>FRS35.Benefit_Unit[] (continued)</i>	1800
<i>FRS35.Benefit_Unit[.QChInt4.Child[]</i>	1802
<i>FRS35.Benefit_Unit[.QChInt4.Child[.QChIntr[]</i>	1803
<i>FRS35.Benefit_Unit[.QChInt4.Child[] (continued)</i>	1806
<i>FRS35.Benefit_Unit[.QChInt4.Child[.QChIntr[]</i>	1807
<i>FRS35.Benefit_Unit[.QChInt4.Child[] (continued)</i>	1810
<i>FRS35.Benefit_Unit[.QChInt4.Child[.QChIntr[]</i>	1811
<i>FRS35.Benefit_Unit[.QChInt4.Child[] (continued)</i>	1814

<i>FRS35.Benefit_Unit[].QChInt4.Child[].QChIntr[]</i>	1815
<i>FRS35.Benefit_Unit[].QChInt4.Child[] (continued)</i>	1818
<i>FRS35.Benefit_Unit[].QChInt4.Child[].QChIntr[]</i>	1819
<i>FRS35.Benefit_Unit[].QChInt4.Child[] (continued)</i>	1822
<i>FRS35.Benefit_Unit[].QChInt4.Child[].QChIntr[]</i>	1823
<i>FRS35.Benefit_Unit[].QChInt4.Child[] (continued)</i>	1826
<i>FRS35.Benefit_Unit[].QChInt4.Child[].QChIntr[]</i>	1827
<i>FRS35.Benefit_Unit[].QChInt4.Child[] (continued)</i>	1830
<i>FRS35.Benefit_Unit[].QChInt4.Child[].QChIntr[]</i>	1831
<i>FRS35.Benefit_Unit[].QChInt5</i>	1834
<i>FRS35.Benefit_Unit[].QChInt5.Child[]</i>	1838
<i>FRS35.Benefit_Unit[] (continued)</i>	1839
<i>FRS35.Benefit_Unit[].QNS.Ad[]</i>	1846
<i>FRS35.Benefit_Unit[].QNS.Ad[].QNSAmt[]</i>	1849
<i>FRS35.Benefit_Unit[].QNS.Ad[] (continued)</i>	1851
<i>FRS35.Benefit_Unit[].QCNS</i>	1852
<i>FRS35.Benefit_Unit[].QCNS.Ch[]</i>	1853
<i>FRS35.Benefit_Unit[].QCNS.Ch[].QNSAmt[]</i>	1857
<i>FRS35.Benefit_Unit[].QCNS.Ch[] (continued)</i>	1859
<i>FRS35.Benefit_Unit[] (continued)</i>	1860
<i>FRS35.Benefit_Unit[].QEnd</i>	1862
<i>FRS35.Benefit_Unit[] (continued)</i>	1868
<i>FRS35 (continued)</i>	1869
<i>FRS35.Assets</i>	1871
<i>FRS35.Assets.QCurrAC[]</i>	1873
<i>FRS35.Assets.QCurrAC[].Account[]</i>	1874
<i>FRS35.Assets (continued)</i>	1879
<i>FRS35.Assets.QASaveAC[]</i>	1880
<i>FRS35.Assets.QASaveAC[].Account[]</i>	1881
<i>FRS35.Assets (continued)</i>	1889
<i>FRS35.Assets.QAEquity[]</i>	1890
<i>FRS35.Assets.QAEquity[].Equity[]</i>	1891
<i>FRS35.Assets.QAEquity[] (continued)</i>	1901
<i>FRS35.Assets (continued)</i>	1902
<i>FRS35.Assets.QACertif[]</i>	1903
<i>FRS35.Assets.QACertif[].Certif[]</i>	1904

<i>FRS35.Assets.QACertif[] (continued)</i>	1912
<i>FRS35.Assets (continued)</i>	1913
<i>FRS35.Assets.QPGIB[]</i>	1914
<i>FRS35.Assets.QSaye[]</i>	1917
<i>FRS35.Assets.QSaye[].Saye[]</i>	1918
<i>FRS35.Assets.QSaye[] (continued)</i>	1925
<i>FRS35.Assets (continued)</i>	1926
<i>FRS35.Assets.QPremium[]</i>	1927
<i>FRS35.Assets (continued)</i>	1930
<i>FRS35.Assets.QNSIB[]</i>	1931
<i>FRS35.Assets.QABonds[]</i>	1935
<i>FRS35.Assets.QABonds[].Bond[]</i>	1936
<i>FRS35.Assets.QABonds[] (continued)</i>	1943
<i>FRS35.Assets (continued)</i>	1944
<i>FRS35.Assets.QCBonds[]</i>	1945
<i>FRS35.Assets.QCBonds[].Bond[]</i>	1946
<i>FRS35.Assets.QCBonds[] (continued)</i>	1953
<i>FRS35.Assets (continued)</i>	1954
<i>FRS35.Assets.QFirstOp[]</i>	1955
<i>FRS35.Assets (continued)</i>	1958
<i>FRS35.Assets.QYPlan[]</i>	1959
<i>FRS35.Assets (continued)</i>	1962
<i>FRS35.Recall</i>	1969
<i>FRS35.Recall.QRecInt</i>	1970
<i>FRS35.Recall.QRecInt.Person[]</i>	1971
<i>FRS35.Recall.QRecPers.Person[]</i>	1972
<i>FRS35 (continued)</i>	1978
<i>FRS35.Admin</i>	1979
<i>FRS35.Admin.QTOcc</i>	1983
<i>FRS35.Admin.QTOcc.QSOC[]</i>	1984
<i>FRS35.Admin.QTOcc (continued)</i>	1990
<i>FRS35.Admin (continued)</i>	1991
<i>FRS35.Admin.QIndOutc</i>	2004
<i>FRS35.Admin.QIndOutc.P[]</i>	2005
<i>FRS35.Admin (continued)</i>	2007
<i>FRS35.Admin.QCalls.Call[]</i>	2008

<i>FRS35.Admin.QCalls</i> _____	2011
<i>FRS35.Admin (continued)</i> _____	2012
<i>FRS35.Admin.QConfid</i> _____	2013
<i>FRS35.Admin.QConfid.QPersConf</i> _____	2014
<i>FRS35.Admin.QConfid.QPersConf.P[]</i> _____	2016
<i>FRS35.Admin.QConfid.QPersConf (continued)</i> _____	2019
<i>FRS35.Admin.QConfid (continued)</i> _____	2020
<i>FRS35.Admin.QObsSheet</i> _____	2021
<i>FRS35.Admin (continued)</i> _____	2024
<i>FRS35.Admin.QNonResp</i> _____	2025
<i>FRS35.Admin.QNonResp.QIndNonResp</i> _____	2029
<i>FRS35.Admin.QNonResp.QIndNonResp.Adult[]</i> _____	2030
<i>FRS35.Admin.QNonResp.QIndNonResp (continued)</i> _____	2033
<i>FRS35.Admin.QNonResp (continued)</i> _____	2034
<i>FRS35.Admin (continued)</i> _____	2041
<i>FRS35.Admin.MF</i> _____	2045
<i>FRS35 (continued)</i> _____	2046



## 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 :**

## **SSTRReg**

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.SSTRReg := DBData.QDataBag.SSTRReg**

---

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 Hholder?

- (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]

```

```

StilleEduc := (' - 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])

```

```

StilleEduc := ''

```

---

```

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: HHSsize > 0
  AND: P[FHHSsize].Sex = RESPONSE
  AND: In loop FOR P1 := 1 TO FHHSsize
  AND: In loop FOR P2 := 1 TO FHHSsize
  AND: P[P1].QRel[P2].R IN [Child .. FChild]
  AND: P[P1].Parent1 = EMPTY

```

**P[P1].Parent1 := P2**

---

```

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

```

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

---

```

WARN IF: HHSsize > 0
  AND: P[FHHSsize].Sex = RESPONSE
  AND: In loop FOR P1 := 1 TO FHHSsize
  AND: In loop FOR P2 := 1 TO FHHSsize
  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: HHSsize > 0
  AND: P[FHHSsize].Sex = RESPONSE
  AND: In loop FOR P1 := 1 TO FHHSsize
  AND: In loop FOR P2 := 1 TO FHHSsize
  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

**PPrec[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

**PPrec[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

**PPrec[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

**PPrec[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)

```

**PRC[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)

```

**PRC[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

**Pre1.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]

**HoHPtrnr := 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  $\epsilon^{\wedge}$ RentWkly a week.

Rents for Council tenants are normally below  $\epsilon$ 100 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 < RMort**

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:** (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = Two) AND (i = 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:** (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = Two) AND (i = 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 æ^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 (æ^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 (€' + 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: (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = Two)
AND (i = 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: (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = Two)
AND (i = 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:** (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = Two) AND (i = 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:** (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = Two) AND (i = 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:** (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = 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:** (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = 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 REPAYED 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: (((PPurcLoan IN [One .. Two]) AND (i = 1)) OR ((PPurcLoan = 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<sup>^</sup>last premium<sup>^</sup>included for this<sup>^</sup>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.