```
unit LCRBenefits;

interface

uses Windows, SysUtils, Classes,
     LCRConfig, LCRPWSRecords, LCRCosts, LCRGlobals,
     LCRMetricCollector, LCRResultsFile, SafewaterUncertBucket,
     Math, CodeSiteLogging, Generics.Collections;

type
  TBenType = (btVoluntary, btMandatory, btRequested);

  TLCRBenYears = class
    Year : array[1..150] of double;
  end;

  TLCRBenByYear = class
    fTotYears : integer;
    fOutName  : string;
    fBenYears : TObjectDictionary<string,TLCRBenYears>;
    Active : boolean;

    constructor create(aConfig : TLCRConfig; aName : string);
    procedure AddBen(aName : string);
    procedure UpdateBen(Yr : integer; const aName : string; aValue : double);
    procedure SaveResults;
    destructor Destroy; override;
  end;

  TLCRBenefits=class
  private
    fConfig : TLCRConfig;
    fOutputs : TMetricList;
    fDummyProb : double;
    fYearsOfOutput : integer;
    fDebug : TBufferedFileStream;
    fDebug2 : TBufferedFileStream;
    fDebugTemp,fDebugTemp2 : TStringList;
    fPWSID:string;
    fBBYY : TLCRBenByYear;

    dYr,dA,dS : integer;
    bp1, bp2: integer;

    function CorrectBL(const BL: double; const Age,Sex: integer) : double;
    function FCVD(const BL1, BL2: double; const Age, Sex: integer): double;
    function FADHD(const BL1, BL2: double): double;
    function FLBW(const BL1, BL2: double): double;
```

```
    procedure D(const S : string);
    procedure DWrite;
    procedure D2(const S: string);
    procedure DWrite2;
    procedure DWrite3(s: string);
  public
    DummyProb,BenPop,Ben7,BenA,Ben11,Ben0 : double;
    IQLossVal,CVDVal,ADHDVal,LBWVal, TotBen,
    IQLoss_C_BP1, IQLossVal_C_BP1,
    IQLoss_C_BP2, IQLossVal_C_BP2,
    CVD_C, CVDVal_C, ADHDVal_C,ADHD_C, LBWVal_C,LBW_C : double;
    IQLoss_L_Vol, IQLossVal_L_Vol,
    IQLoss_L_Mand, IQLossVal_L_Mand,
    IQLoss_L_Req, IQLossVal_L_Req,
    IQLoss_POU, IQLossVal_POU,
    CVD_L, CVDVal_L, ADHDVal_L,ADHD_L, LBWVal_L,LBW_L : double;

    //PBBlood by sex,age,bin
    BL : array[1..2,0..80,1..16] of double;
    CVDRate : array[1..2,4..8] of double;
    VSL,IQPointVal,ADHDCaseVal : double;

    FinalBins : array[1..16] of double;
    DoDebugOUt : boolean;


    UncertaintyVars : TUncertaintyStudy;  //pointer to model level var - must be set

    function FIQLoss(const BL1,BL2 : double) : double;
    procedure CalcBinMove(const FromBin,ToBin,Yr : integer; const Pop : double;
const CCT : boolean; BenType : TBenType; Proxy : boolean; AdhocDebug:
boolean=false);

    constructor create(aConfig : TLCRConfig; aOutputMetrics : TMetricList;
Uncertainty : TUncertaintyStudy;
                       aOption: string);
    destructor Destroy; override;

    procedure CalcBenefitsNew(const A : TYearlyMovementMicro; const Wgt : double;
const DoDebug: boolean; const aCosts : TLCRCosts;
          const P90CCT, P90LSL : array of double;  abp1,abp2 : integer; SmallSystem
: boolean; ProxySystem : boolean);

  end;

  TBenefitsCollector=class
  private
```

```
    fConfig: TLCRConfig;
    fOutputs: TMetricList;
    fUncertainty: TUncertaintyStudy;
    fDummyProb: double;

    BenefitsBaseline: TLCRBenefits;
    BenefitsOption: TLCRBenefits;

    bp1, bp2: integer;
  public
    DummyProb,BenPop,Ben7,BenA,Ben11,Ben0 : double;
    IQLossVal,CVDVal,TotBen,ADHDVal,LBWVal,
    IQLoss_C_BP1, IQLossVal_C_BP1,
    IQLoss_C_BP2, IQLossVal_C_BP2,

    CVD_C, CVDVal_C, ADHDVal_C,ADHD_C, LBWVal_C,LBW_C : double;
    IQLoss_L_Vol, IQLossVal_L_Vol,
    IQLoss_POU, IQLossVal_POU,
    IQLoss_L_Mand, IQLossVal_L_Mand,
    IQLoss_L_Req, IQLossVal_L_Req,
    CVD_L, CVDVal_L, ADHDVal_L,ADHD_L, LBWVal_L,LBW_L : double;

    EndingBins, StartingBins,FinalBins, EndingBinsCheck : array[1..16] of double;
    BinMovements : TMovementMicro;

    DoDebugOut : boolean;
    NewBenBins : boolean;

    procedure GenerateBenefits(aCosts : TLCRCosts; ProxySystem : boolean);
    constructor create(aConfig : TLCRConfig; aOutputMetrics : TMetricList;
aUncertainty : TUncertaintyStudy);
    destructor Destroy; override;
  end;



implementation

var PBF,PBM,PBSens1,PBSens2,PBSens3 : string;
    //just squeezing in blood leads for women of child bearing age here,...
    BLCBA : array[1..16] of double =
(1.75,1.17,0.68,1.31,1.31,1.31,0.96,0.96,0.96,0.68,0.68,0.68,1.06,0.84,0.68,0.68);
    //Income change from 1 iq point change in 2016 dollars) - Matt L email 7/29/2020
    IncomeDeltaAge: array[0..80] of double = (
      0,0,0,0,0,0,0,0,0,0,
      -8,-3,-7,-4,-18,-36,75,53,-1891,-2685,
      874,1704,1405,1974,4243,6387,7682,8524,9431,10206,
      10774,11808,12315,13129,13778,14872,15658,16184,16781,17367,
```

```
      17788,18614,18865,19454,19958,20525,21185,21407,21926,22241,
      22273,22865,22810,23029,22782,22484,22174,21640,21131,20339,
      18978,17958,16313,14295,12830,10713,8979,7425,6329,5370,
      4437,3678,3287,2875,2596,2212,1952,1703,1525,1288,1071
    );

function Discount(const Value : double; const Yrs : integer; const Rate : double) :
double;
begin
   Result:=Value/intpower((1+Rate),Yrs);
end;

function Annualize(const DiscRate,value,Years : double) : double;
begin
  Result:= Value * (DiscRate / (1 - Power((1 + DiscRate),-Years)));
end;

{ TLCRBenefits }

procedure TLCRBenefits.CalcBinMove(const FromBin,ToBin,Yr : integer; const Pop :
double; const CCT : boolean;
                                    BenType : TBenType; Proxy : boolean; AdhocDebug:
boolean=false);
var a,s,y,newage,yold,ynew,ti,fi : integer;
    FBL,TBL,CalcBL,CPop,tmpPop, iFBL, iTBL, FBL0, TBL0,tmpv : double;
    valLBW1,valLBW2 : double;
    IQL0 : double;
    vIQ, vCVD, vADHD, vLBW : double;
    PBWin : integer;
    Dout : boolean;
    Counted40 : boolean;
    CountPop : double;
    StartYear : integer;
begin
  PBWin:=10;
  BenPop:=BenPop+Pop;
  FinalBins[ToBin]:=FinalBins[ToBin]+Pop;

  D('Move from '+FromBin.ToString+' to ' +ToBin.ToString+': '+Pop.ToString);

  //special case to save new benefits for incoming infants...
  FBL:=0; TBL:=0;
  for fi:=0 to 6 do begin
     FBL:=FBL+BL[1,fi,FromBin];
     TBL:=TBL+BL[1,fi,ToBin];
  end;
  FBL0:=FBL/7;
  TBL0:=TBL/7;
```

```
  IQL0:=FIQLoss(FBL0,TBL0);

if AdhocDebug then
DWrite3('Year,Sex,FutureYear,StartAge,CurAge,Pop,FromBin,ToBin,FBL,TBL,BenType,Case,
DiscDollars');

  for s:=1 to 2 do begin
    for a:=0 to 80 do begin
      if (not fConfig.BenefitsAll) and (a>7) then break;

      if (s=2) and ( (a=3) or (a=50) or (a=0)) and (fConfig.Debug) then Dout:=true
else Dout:=false;
      NewAge:=a;
      //years in old regime...
      Yold:=PBWin;

      Counted40:=False;
      //Change to delay CCT moves by 2 years 08/25/20
      StartYear := Yr;
      if CCT then StartYear:=StartYear+2;
      //go through remaining years with this cohort...
      for y:=StartYear to fConfig.YearsOfOutput do begin
        CPop:=Pop*fConfig.DefaultPopPct[s,NewAge];

        //get no averaging blood leads...
        iTBL:=BL[s,newage,ToBin];
        iFBL:=BL[s,newage,FromBin];

        //get average for new condition
        TBL:=0;
        FBL:=0;
        ti:=0;
        //find old vs new blood lead
        for fi:=max(0,a-Yold+1) to a-1 do begin

          TBL:=TBL+BL[s,fi,FromBin];
          FBL:=FBL+BL[s,fi,FromBin];
          inc(ti);
        end;
        //add new component to BL
        fi:=max(a,newage-PBWin);
        while (fi<=newage) do begin
          TBL:=TBL+BL[s,fi,ToBin];
          FBL:=FBL+BL[s,fi,FromBin];
          inc(ti);
          inc(fi);
        end;
```

```
        if ti>0 then begin
          TBL:=TBL/ti;
          FBL:=FBL/ti;
        end else begin
          continue;  // probably an error condition
          Codesite.Send('error newage:',newage);
        end;

        if NewAge=6 then begin
          vIQ:=FIQLoss(FBL,TBL) * CPop;
          if vIQ>0 then begin
            Ben7:=Ben7 + CPop;
          end;
          if CCT then begin
            if not (BenType = btMandatory) then begin
              IQLoss_C_BP1:=IQLoss_C_BP1+vIQ;
              if not Proxy then
                fBBYY.UpdateBen(y,'IQ CCT BP1',vIQ);
              tmpv := Discount(vIQ*IQPointVal,y,fConfig.DiscountRate);
              IQLossVal_C_BP1:=IQLossVal_C_BP1 + tmpv;
            end else begin
              IQLoss_C_BP2:=IQLoss_C_BP2+vIQ;
              if not Proxy then
                fBBYY.UpdateBen(y,'IQ CCT BP2',vIQ);
              tmpv := Discount(vIQ*IQPointVal,y,fConfig.DiscountRate);
              IQLossVal_C_BP2:=IQLossVal_C_BP2 + tmpv;
            end;
          end else begin
            if ToBin = 16 then begin
              IQLoss_POU:=IQLoss_POU+vIQ;
              if not Proxy then
                fBBYY.UpdateBen(y,'IQ POU',vIQ);
              tmpv := Discount(vIQ*IQPointVal,y,fConfig.DiscountRate);
              IQLossVal_POU:=IQLossVal_POU + tmpv;
            end else begin
              case BenType of
                 btVoluntary : begin
                                 IQLoss_L_Vol:=IQLoss_L_Vol+vIQ;
                                 if not Proxy then
                                   fBBYY.UpdateBen(y,'IQ LSLR Vol',vIQ);
                                 tmpv :=
Discount(vIQ*IQPointVal,y,fConfig.DiscountRate);
                                 IQLossVal_L_Vol:=IQLossVal_L_Vol + tmpv;
                               end;
                 btMandatory : begin
                                 IQLoss_L_Mand:=IQLoss_L_Mand+vIQ;
                                 if not Proxy then
                                   fBBYY.UpdateBen(y,'IQ LSLR Mand',vIQ);
```

```
                                tmpv :=
Discount(vIQ*IQPointVal,y,fConfig.DiscountRate);
                                IQLossVal_L_Mand:=IQLossVal_L_Mand + tmpv;
                              end;
                btRequested : begin
                                IQLoss_L_Req:=IQLoss_L_Req+vIQ;
                                if not Proxy then
                                  fBBYY.UpdateBen(y,'IQ LSLR Req',vIQ);
                                tmpv :=
Discount(vIQ*IQPointVal,y,fConfig.DiscountRate);
                                IQLossVal_L_Req:=IQLossVal_L_Req + tmpv;
                              end;
              end;
            end;
          end;
        end;

        //low birth weight
        if (a=0) and (y>StartYear) then begin
          //counting newborns in every year of analysis. After first year
          tmpPop:=Pop*fConfig.DefaultPopPct[s,0] ;

          vLBW:=FLBW(BLCBA[FromBin],BLCBA[ToBin]) / 20;
          //leaving this like this so it is clear what is going on....
          valLBW1:= 1519.3*vLBW*0.3/100*tmpPop + //2.5
                    1139.26*vLBW*0.3/100*tmpPop + //3
                    958.54*vLBW*0.5/100*tmpPop +  //3.3
                    640.49*vLBW*0.9/100*tmpPop +  //4
                    480.36*vLBW*1.3/100*tmpPop +  //4.5
                    360.2*vLBW*2.4/100*tmpPop +  //5
                    14.86*vLBW*4.1/100*tmpPop +   //5.5
                    14.61*vLBW*13.5/100*tmpPop +   //6
                    14.12*vLBW*33.2/100*tmpPop +   //7
                    13.66*vLBW*29.4/100*tmpPop ;   //8

          valLBW2:= 0.00*vLBW*0.3/100*tmpPop + //2.5
                    593.17*vLBW*0.3/100*tmpPop + //3
                    469.72*vLBW*0.5/100*tmpPop +  //3.3
                    271.85*vLBW*0.9/100*tmpPop +  //4
                    183.57*vLBW*1.3/100*tmpPop +  //4.5
                    123.74*vLBW*2.4/100*tmpPop +  //5
                    16.04*vLBW*4.1/100*tmpPop +   //5.5
                    14.34*vLBW*13.5/100*tmpPop +   //6
                    11.45*vLBW*33.2/100*tmpPop +   //7
                    9.12*vLBW*29.4/100*tmpPop ;   //8

          if vLBW>0 then begin
            Ben0:=Ben0 + tmpPop;
```

```
          end;
          if CCT then begin
            LBW_C:=LBW_C+(vLBW * 20 * tmpPop) ;
            LBWVal_c:=LBWVal_c + Discount((valLBW1 +
2*valLBW2),y,fConfig.DiscountRate);
          end else begin
            LBW_L:=LBW_L+(vLBW * 20 * tmpPop);
            LBWVal_L:=LBWVal_l + Discount((valLBW1 +
2*valLBW2),y,fConfig.DiscountRate);
          end;
        end;


        if (y>StartYear) and (a=0) and (y<=fConfig.YearsOfOutput-6) then begin
          //add on new births that will receive benefits...
          vIQ:=IQL0 * Pop*fConfig.DefaultPopPct[s,6];
          if vIQ>0 then begin
            Ben7:=Ben7+Pop*fConfig.DefaultPopPct[s,6];
          end;

          if CCT then begin
            if not (BenType = btMandatory) then begin
              IQLoss_C_BP1:=IQLoss_C_BP1+vIQ;
              if not Proxy then
                fBBYY.UpdateBen(y+6,'IQ CCT BP1',vIQ);
              tmpv := Discount(vIQ*IQPointVal,y+6,fConfig.DiscountRate);
              IQLossVal_C_BP1:=IQLossVal_C_BP1 + tmpv;
            end else begin
              IQLoss_C_BP2:=IQLoss_C_BP2+vIQ;
              if not Proxy then
                fBBYY.UpdateBen(y+6,'IQ CCT BP2',vIQ);
              tmpv := Discount(vIQ*IQPointVal,y+6,fConfig.DiscountRate);
              IQLossVal_C_BP2:=IQLossVal_C_BP2 + tmpv;
            end;
          end else begin
            if ToBin = 16 then begin
              IQLoss_POU:=IQLoss_POU+vIQ;
              if not Proxy then
                fBBYY.UpdateBen(y+6,'IQ POU',vIQ);
              tmpv := Discount(vIQ*IQPointVal,y+6,fConfig.DiscountRate);
              IQLossVal_POU:=IQLossVal_POU + tmpv;
            end else begin
              case BenType of
                 btVoluntary : begin
                                 IQLoss_L_Vol:=IQLoss_L_Vol+vIQ;
                                 if not Proxy then
                                   fBBYY.UpdateBen(y+6,'IQ LSLR Vol',vIQ);
                                 tmpv :=
```

```
Discount(vIQ*IQPointVal,y+6,fConfig.DiscountRate);
                                IQLossVal_L_Vol:=IQLossVal_L_Vol + tmpv;
                              end;
                btMandatory : begin
                                IQLoss_L_Mand:=IQLoss_L_Mand+vIQ;
                                if not Proxy then
                                  fBBYY.UpdateBen(y+6,'IQ LSLR Mand',vIQ);
                                tmpv :=
Discount(vIQ*IQPointVal,y+6,fConfig.DiscountRate);
                                IQLossVal_L_Mand:=IQLossVal_L_Mand + tmpv;
                              end;
                btRequested : begin
                                IQLoss_L_Req:=IQLoss_L_Req+vIQ;
                                if not Proxy then
                                  fBBYY.UpdateBen(y+6,'IQ LSLR Req',vIQ);
                                tmpv :=
Discount(vIQ*IQPointVal,y+6,fConfig.DiscountRate);
                                IQLossVal_L_Req:=IQLossVal_L_Req + tmpv;
                              end;
              end;
            end;
          end;

if AdhocDebug then
DWrite3(StartYear.ToString+','+s.ToString+','+y.ToString+','+a.ToString+','+

newage.ToString+','+(Pop*fConfig.DefaultPopPct[s,6]).tostring+','+FromBin.ToString+'
,' +ToBin.ToString+','+
  FBL0.ToString+','+TBL0.ToString +',IQI,'+viq.ToString+','+tmpv.ToString);

        end;

        if NewAge>=40 then begin
          if fConfig.RunNoBLAveraging then
            vCVD:=fCVD(iFBL,iTBL,NewAge,S) * CPop
          else
            vCVD:=fCVD(FBL,TBL,NewAge,S) * CPop;
          //count this cohort pop once
          CountPop:=0;
          if (not Counted40) and (vCVD>0) then begin
            BenA:=BenA+CPop;
            Counted40:=True;
            CountPop:=CPop;
          end;

          if CCT then begin
            CVD_C:=CVD_C+vCVD;
            CVDVal_C:=CVDVal_C+Discount(vCVD*VSL,y,fConfig.DiscountRate);
```

```
          end else begin
            CVD_L:=CVD_L+vCVD;
            CVDVal_L:=CVDVal_L+Discount(vCVD*VSL,y,fConfig.DiscountRate);
          end;

          if Dout then begin

D(',,,CVD,'+fCVD(FBL,TBL,NewAge,S).ToString+',PopTotal,'+vCVD.ToString+',Value,'+Dis
count(vCVD*VSL,y,fConfig.DiscountRate).ToString);
          end;

        end;

        NewAge:=NewAge+1;
        Dec(YOld);
        if NewAge>80 then break;
      end;
    end;
  end;
end;

function TLCRBenefits.CorrectBL(const BL: double; const Age,Sex: integer): double;
begin
  if BL<0.76 then begin
    if trunc(Age/10)=4 then begin
      if Sex=1 then Result:=0.92 else Result:=1.07;
    end else
    if trunc(Age/10)=5 then begin
      if Sex=1 then Result:=0.97 else Result:=1.05;
    end else
    if trunc(Age/10)=6 then begin
      if Sex=1 then Result:=0.99 else Result:=1.05;
    end else
    if trunc(Age/10)=7 then begin
      if Sex=1 then Result:=1.03 else Result:=1.04;
    end else
      if Sex=1 then Result:=1.03 else Result:=1.04;
  end else

  if BL<1.12 then begin
    if trunc(Age/10)=4 then begin
      if Sex=1 then Result:=0.93 else Result:=1.04;
    end else
    if trunc(Age/10)=5 then begin
      if Sex=1 then Result:=0.96 else Result:=1.04;
    end else
    if trunc(Age/10)=6 then begin
      if Sex=1 then Result:=0.94 else Result:=1.02;
```

```
    end else
    if trunc(Age/10)=7 then begin
      if Sex=1 then Result:=0.98 else Result:=1.05;
    end else
      if Sex=1 then Result:=0.98 else Result:=1.05;
  end else

  if BL<1.71 then begin
    if trunc(Age/10)=4 then begin
      if Sex=1 then Result:=0.93 else Result:=1.05;
    end else
    if trunc(Age/10)=5 then begin
      if Sex=1 then Result:=0.96 else Result:=1.02;
    end else
    if trunc(Age/10)=6 then begin
      if Sex=1 then Result:=0.95 else Result:=1.03;
    end else
    if trunc(Age/10)=7 then begin
      if Sex=1 then Result:=0.99 else Result:=1.05;
    end else
      if Sex=1 then Result:=0.99 else Result:=1.05;
  end else

  begin
    if trunc(Age/10)=4 then begin
      if Sex=1 then Result:=0.94 else Result:=1.05;
    end else
    if trunc(Age/10)=5 then begin
      if Sex=1 then Result:=0.94 else Result:=1.01;
    end else
    if trunc(Age/10)=6 then begin
      if Sex=1 then Result:=0.95 else Result:=1.04;
    end else
    if trunc(Age/10)=7 then begin
      if Sex=1 then Result:=0.98 else Result:=1.04;
    end else
      if Sex=1 then Result:=0.98 else Result:=1.04;
  end;

  Result:=Result*BL;
end;

procedure TLCRBenefits.CalcBenefitsNew(const A : TYearlyMovementMicro; const Wgt :
double; const DoDebug: boolean; const aCosts : TLCRCosts;
          const P90CCT, P90LSL : array of double;  abp1,abp2 : integer; SmallSystem
: boolean; ProxySystem : boolean);
var Y,F,T : integer;
    DoCCT : boolean;
```

```
    BenType : TBenType;
begin
  bp1:=abp1;
  bp2:=abp2;

  IQLoss_C_BP1:=0; IQLossVal_C_BP1:=0;
  IQLoss_C_BP2:=0; IQLossVal_C_BP2:=0;
  CVD_C:=0; CVDVal_C:=0;
  ADHD_C:=0; ADHDVal_C:=0;
  LBW_C:=0; LBWVal_C:=0;
  IQLoss_L_Vol:=0; IQLossVal_L_Vol:=0;
  IQLoss_POU:=0; IQLossVal_POU:=0;
  IQLoss_L_Mand:=0; IQLossVal_L_Mand:=0;
  IQLoss_L_Req:=0; IQLossVal_L_Req:=0;
  CVD_L:=0; CVDVal_L:=0;
  ADHD_L:=0; ADHDVal_L:=0;
  LBW_L:=0; LBWVal_L:=0;
  DoDebugOUt:=DoDebug;

  BenPop:=0; Ben7:=0; BenA:=0; Ben0:=0; Ben11:=0;
  fillchar(FinalBins,sizeof(FinalBins),0);
  fDebugTemp.Clear;
  fDebugTemp2.Clear;
  //dump more costing data if necc....
  D('Starting PWS:'+aCosts.CostingData.PWSid);
  fPWSID:=aCosts.CostingData.PWSid;

  for y:=0 to fConfig.YearsOfOutput do begin
    if y=0 then begin
      continue;
    end;
    D('Starting Year:'+y.ToString);



    for f:=low(A[y]) to high(A[y]) do begin
      for t:=low(A[y,f]) to high(A[y,f]) do begin
        if A[y,f,t]=0 then continue;
        DoCCT:=True;
        if GMoveBinLC[f,t]>1 then DoCCT:=False;
        BenType := btMandatory;
        if DoCCT then begin
          if (p90CCT[y] <= bp2) then
            BenType := btVoluntary;
        end else begin
          if SmallSystem then begin
            if (p90LSL[y] <= bp2) then
               BenType := btRequested;
```

```
          end else begin
            if (p90LSL[y] <= bp1) then
              BenType := btRequested
            else
            if (p90LSL[y] <= bp2) then
              BenType := btVoluntary;
          end;
        end;
        D2(fPWSID);
        CalcBinMove(f,t,Y,A[y,f,t] * Wgt,DoCCT, BenType, ProxySystem);
        D2('');
      end;
    end;
  end;



IQLossVal_C_BP1:=Annualize(fConfig.DiscountRate,IQLossVal_C_BP1,fConfig.YearsOfOutpu
t);

IQLossVal_C_BP2:=Annualize(fConfig.DiscountRate,IQLossVal_C_BP2,fConfig.YearsOfOutpu
t);
  CVDVal_C:=Annualize(fConfig.DiscountRate,CVDVal_C,fConfig.YearsOfOutput);
  ADHDVal_C:=Annualize(fConfig.DiscountRate,ADHDVal_C,fConfig.YearsOfOutput);
  LBWVal_C:=Annualize(fConfig.DiscountRate,LBWVal_C,fConfig.YearsOfOutput);
  CVD_C:=CVD_C/fConfig.YearsOfOutput;
  IQLoss_C_BP1:=IQLoss_C_BP1/fConfig.YearsOfOutput;
  IQLoss_C_BP2:=IQLoss_C_BP2/fConfig.YearsOfOutput;
  ADHD_C:=ADHD_C/fConfig.YearsOfOutput;
  LBW_C:=LBW_C/fConfig.YearsOfOutput;


IQLossVal_L_Vol:=Annualize(fConfig.DiscountRate,IQLossVal_L_Vol,fConfig.YearsOfOutpu
t);

IQLossVal_L_Mand:=Annualize(fConfig.DiscountRate,IQLossVal_L_Mand,fConfig.YearsOfOut
put);

IQLossVal_L_Req:=Annualize(fConfig.DiscountRate,IQLossVal_L_Req,fConfig.YearsOfOutpu
t);

IQLossVal_POU:=Annualize(fConfig.DiscountRate,IQLossVal_POU,fConfig.YearsOfOutput);
  CVDVal_L:=Annualize(fConfig.DiscountRate,CVDVal_L,fConfig.YearsOfOutput);
  ADHDVal_L:=Annualize(fConfig.DiscountRate,ADHDVal_L,fConfig.YearsOfOutput);
  LBWVal_L:=Annualize(fConfig.DiscountRate,LBWVal_L,fConfig.YearsOfOutput);
  CVD_L:=CVD_L/fConfig.YearsOfOutput;
  IQLoss_L_Vol:=IQLoss_L_Vol/fConfig.YearsOfOutput;
  IQLoss_L_Mand:=IQLoss_L_Mand/fConfig.YearsOfOutput;
```

```
  IQLoss_L_Req:=IQLoss_L_Req/fConfig.YearsOfOutput;
  IQLoss_POU:=IQLoss_POU/fConfig.YearsOfOutput;
  ADHD_L:=ADHD_L/fConfig.YearsOfOutput;
  LBW_L:=LBW_L/fConfig.YearsOfOutput;

  IQLossVal:=IQLossVal_C_BP1+IQLossVal_L_Vol+IQLossVal_C_BP2+IQLossVal_L_Mand+
IQLossVal_L_Req+
             IQLossVal_POU;

  CVDVal:=CVDVal_C+CVDVal_L;
  ADHDVal:=ADHDVal_C+ADHDVal_L;
  LBWVal:=LBWVal_C+LBWVal_L;

  //TotBen:=IQLossVal+CVDVal+ADHDVal+LBWVal;
  // don't include adhd and low birth weight
  TotBen:=IQLossVal+CVDVal;

  if (BenPop>0) and (Ben7=0) and (BenA=0) then begin
    fDebugTemp.Clear;
    D('Starting PWS:'+aCosts.CostingData.PWSid);
    D(',Bin movement:'+floattostr(BenPop)+ '  no BL changes');
    DWrite;
  end;
  if Ben7+BenA>0 then begin
    DWrite;
    DWrite2;
  end;

end;


function RCB(const c : integer) : integer;
//this converts the BL table numbering to the Bin Numbering in pops.
begin
  // bin numbering has been corrected
  result:=c;
  exit;

  if c=4 then result:=1 else
  if c=7 then result:=2 else
  if c=1 then result:=3 else
  if c=5 then result:=4 else
  if c=8 then result:=5 else
  if c=2 then result:=6 else
  if c=6 then result:=7 else
  if c=9 then result:=8 else
  if c=3 then result:=9 else
    result:=c;
```

```
end;


constructor TLCRBenefits.create;
var i,j,cMCL,cDR : integer;
    T,TL : TStringLIst;
    a : integer;
    v : double;
begin
  fConfig:=aConfig;
  fOutputs:=aOutputMetrics;
  fDummyProb:=1;
  DoDebugOUt:=False;
  fBBYY := TLCRBenByYear.create(aConfig, aOption);
  fBBYY.AddBen('IQ CCT BP1');
  fBBYY.AddBen('IQ CCT BP2');
  fBBYY.AddBen('IQ LSLR Vol');
  fBBYY.AddBen('IQ LSLR Mand');
  fBBYY.AddBen('IQ LSLR Req');
  fBBYY.AddBen('IQ POU');

  fYearsOfOutput := fConfig.YearsOfOutput;
  UncertaintyVars:=Uncertainty;


  fillchar(BL,sizeof(BL),0);
  T:=TstringList.Create;
  TL:=TStringList.Create;

  T.Text:=PBF;
  for i:=0 to T.Count-1 do begin
    TL.CommaText:=T[i];
    for j:=1 to 16 do begin
      BL[2,TL[0].ToInteger,RCB(j)]:=TL[j].ToDouble;
    end;
  end;

  T.Text:=PBM;
  for i:=0 to T.Count-1 do begin
    TL.CommaText:=T[i];

    for j:=1 to 16 do begin
      BL[1,TL[0].ToInteger,RCB(j)]:=TL[j].ToDouble;
    end;
  end;

  //just force this sensitivity stuff in here... getting out of control...
  if fConfig.ChildBLSens>0 then begin
```

```
  if fConfig.ChildBLSens = 1 then T.Text:=PBSens1 else
  if fConfig.ChildBLSens = 2 then T.Text:=PBSens2 else
     T.Text:=PBSens3;
  for i:=0 to 6 do begin
    TL.CommaText:=T[i];
    for j:=1 to 16 do begin
      BL[1,TL[0].ToInteger,RCB(j)]:=TL[j].ToDouble;
      BL[2,TL[0].ToInteger,RCB(j)]:=TL[j].ToDouble;
    end;
  end;
end;


TL.Free;
T.Free;

VSL:=10210000;
if fConfig.IQValueSens=1 then begin
  if Round(fConfig.DiscountRate*100)/100=0.03 then
    IQPointVal:=12155
  else
    IQPointVal:=2977;
end else begin
  if Round(fConfig.DiscountRate*100)/100=0.03 then
    IQPointVal:=22503
  else
    IQPointVal:=5708;
end;

if Round(fConfig.DiscountRate*100)/100=0.03 then
  ADHDCaseVal:=232128
else
  ADHDCaseVal:=108472;

CVDRate[1,4]:=0.00078592;
CVDRate[1,5]:=0.00218595;
CVDRate[1,6]:=0.00459819;
CVDRate[1,7]:=0.01080168;
CVDRate[1,8]:=0.01080168;

CVDRate[2,4]:=0.00037709;
CVDRate[2,5]:=0.00097204;
CVDRate[2,6]:= 0.00221088;
CVDRate[2,7]:=0.00675097;
CVDRate[2,8]:=0.00675097;


fDebug:=TBufferedFileStream.Create(UserPath+fConfig.RunName + '_' + aOption +
```

```
'_BenDebug.csv', fmCreate, 4096);
  fDebugTemp:=TstringLIst.Create;
  fDebug2:=TBufferedFileStream.Create(UserPath+fConfig.RunName + '_' + aOption +
'_BenDebug2.csv', fmCreate, 4096);
  fDebugTemp2:=TstringLIst.Create;
  DoDebugOUt:=True;

D2('PWISID,PWSPop,MoveYear,OnYear,FromBin,ToBin,Sex,StartAge,CurAge,CohortPop,FromBL
,ToBL,BaseRate,CVDCases,CountPop');
  DWrite2;
  DoDebugOUt:=False;

end;


procedure TLCRBenefits.D(const S: string);
var ss : string;
begin
  if (fConfig.Debug) and (DoDebugOut) then begin
    fDebugTemp.Add(s);
  end;
end;

procedure TLCRBenefits.DWrite;
var ss : string;
    i : integer;
begin
  if (fConfig.Debug) and (DoDebugOut) then begin
    for i:=0 to fDebugTemp.Count-1 do begin
      ss:=fDebugTemp[i]+#13#10;
      fDebug.WriteBuffer(ss[1], Length(ss)*SizeOf(Char));
    end;
  end;
end;

procedure TLCRBenefits.D2(const S: string);
var ss : string;
begin
  if (fConfig.Debug) and (DoDebugOut) then begin
    fDebugTemp2.Add(s);
  end;
end;

procedure TLCRBenefits.DWrite2;
var ss : string;
    i : integer;
begin
  if (fConfig.Debug) and (DoDebugOut) then begin
```

```
    for i:=0 to fDebugTemp2.Count-1 do begin
      ss:=fDebugTemp2[i]+#13#10;
      fDebug2.WriteBuffer(ss[1], Length(ss)*SizeOf(Char));
    end;
  end;
end;

procedure TLCRBenefits.DWrite3(s: string);
var ss : string;
begin
  ss:=s+#13#10;
  fDebug.WriteBuffer(ss[1], Length(ss)*SizeOf(Char));
end;


destructor TLCRBenefits.Destroy;
begin
  fDebug.Free;
  fDebugTemp.Free;
  fDebug2.Free;
  fDebugTemp2.Free;
  fBBYY.free;
  inherited;
end;

function TLCRBenefits.FIQLoss(const BL1, BL2: double)  : double;
begin
  if fConfig.IQDRSens=1 then begin
    //Updated from 9/18/20 email to format things closer to the doc equation desc
    if (BL1>=1.47) and (BL2>=1.47) then begin
      Result := 3.14 * ln (BL1/BL2);
    end else
    if (BL1>=1.47) then begin
      Result := 3.14 * ln (BL1/1.47) + (2.1 * (1.47 - BL2));
    end else
      Result :=  2.1 * (BL1 - BL2);
  end else
  if fConfig.IQDRSens=2 then begin
    Result := - 3.14 * ln( BL2 / BL1 );
  end else
    Result := 3.25 * ln( (BL1+1) / (BL2+1) );
end;

function TLCRBenefits.FLBW(const BL1, BL2: double): double;
begin
  Result:=-27.4 * (power(BL2,0.5) - power(BL1,0.5));
end;
```

```
function TLCRBenefits.FADHD(const BL1, BL2: double) : double;
var Brate,Beta,v : double;
begin
  Brate:=0.087;
  Beta:=0.59;
  v:= Beta * (ln(BL1) - ln(BL2));
  Result := Brate - ( BRate / ( (1-BRate) * exp(-v) + BRate ) );
  Result:=Result*-1;
end;

function TLCRBenefits.FCVD(const BL1, BL2 : double; const Age,Sex : integer)  :
double;
var P1,P2 : double;
begin
  P1:=BL1;
  P2:=BL2;
  if P2>P1 then begin
    //CSL('Pb Higher: A:'+Age.ToString+' S:'+Sex.ToString);
  end;
  if fConfig.CVDDRSens=1 then
    Result:=CVDRate[Sex,Age DIV 10] * ( 1 - exp(0.94 * log10(p2/p1)))
  else
    Result:=CVDRate[Sex,Age DIV 10] * ( 1 - exp(0.36 * log10(p2/p1)));
end;

{ TBenefitsCollector }

constructor TBenefitsCollector.create(aConfig: TLCRConfig;
  aOutputMetrics: TMetricList; aUncertainty: TUncertaintyStudy);
var
  i,f,t : integer;
begin
  fConfig := aConfig;
  fOutputs := aOutputMetrics;
  fUncertainty := aUncertainty;
  fDummyProb := 1;
  DoDebugOut:=False;
  NewBenBins:=False;

  if fConfig.RunDifference then
  begin
    BenefitsBaseline := TLCRBenefits.create(fConfig, fOutputs, fUncertainty,
'Baseline');
    BenefitsOption := TLCRBenefits.create(fConfig, fOutputs, fUncertainty,
fConfig.OptionName);
  end
  else if fConfig.RunBaselineOnly then
    BenefitsBaseline := TLCRBenefits.create(fConfig, fOutputs, fUncertainty,
```

```
'Baseline')
  else
    BenefitsOption := TLCRBenefits.create(fConfig, fOutputs, fUncertainty,
fConfig.OptionName);

  for f:=low(BinMovements) to high(BinMovements) do begin
    for t:=low(BinMovements[f]) to high(BinMovements[f]) do begin
      if GMoveBinLC[f,t]>0 then begin
        fOutputs.AddOutputMetric(@BinMovements[f,t],@DummyProb,nil,

'BinMove'+inttostr(f)+'To'+inttostr(t),mtBenefitCounts,False,False,False,fConfig.Opt
ionName,0,true);
      end;
    end;
  end;

  fOutputs.AddOutputMetric(@BenPop,@DummyProb,nil,

'BenefitsBinPopMove',mtBenefitCounts,False,False,False,fConfig.OptionName,0,true);

  fOutputs.AddOutputMetric(@BenA,@DummyProb,nil,

'BenefittingAdults',mtBenefitCounts,False,False,False,fConfig.OptionName,0,true);

  fOutputs.AddOutputMetric(@Ben7,@DummyProb,nil,

'Benefitting7',mtBenefitCounts,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@Ben0,@DummyProb,nil,

'Benefitting0',mtBenefitCounts,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@Ben11,@DummyProb,nil,

'Benefitting11',mtBenefitCounts,False,False,False,fConfig.OptionName,0,true);

  fOutputs.AddOutputMetric(@IQLoss_C_BP1,@DummyProb,nil,
         'IQ Point
Cases_CCT_BP1',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@IQLossVal_C_BP1,@DummyProb,nil,
        'IQ Point
Annual_CCT_BP1',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.Discou
ntRate,true);
  fOutputs.AddOutputMetric(@IQLoss_L_Vol,@DummyProb,nil,
         'IQ Point
Cases_LSLR_Voluntary',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@IQLossVal_L_Vol,@DummyProb,nil,
        'IQ Point
Annual_LSLR_Voluntary',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig
.DiscountRate,true);
```

```
  fOutputs.AddOutputMetric(@IQLossVal_POU,@DummyProb,nil,
        'IQ Point
Annual_POU',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRa
te,true);

  fOutputs.AddOutputMetric(@IQLoss_C_BP2,@DummyProb,nil,
         'IQ Point
Cases_CCT_BP2',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@IQLossVal_C_BP2,@DummyProb,nil,
        'IQ Point
Annual_CCT_BP2',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.Discou
ntRate,true);
  fOutputs.AddOutputMetric(@IQLoss_L_Mand,@DummyProb,nil,
         'IQ Point
Cases_LSLR_Mandatory',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@IQLossVal_L_Mand,@DummyProb,nil,
        'IQ Point
Annual_LSLR_Mandatory',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig
.DiscountRate,true);
  fOutputs.AddOutputMetric(@IQLoss_L_Req,@DummyProb,nil,
         'IQ Point
Cases_LSLR_Requested',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@IQLossVal_L_Req,@DummyProb,nil,
        'IQ Point
Annual_LSLR_Requested',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig
.DiscountRate,true);
  fOutputs.AddOutputMetric(@IQLoss_POU,@DummyProb,nil,
        'IQ Point
Cases_POU',mtBenefitCases,False,False,False,fConfig.OptionName,fConfig.DiscountRate,
true);


  fOutputs.AddOutputMetric(@IQLossVal,@DummyProb,nil,
        'IQ Point
Annual',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRate,t
rue);


  fOutputs.AddOutputMetric(@CVD_C,@DummyProb,nil,
         'CVD Annual
Cases_CCT',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@CVDVal_C,@DummyProb,nil,
        'CVD
Annual_CCT',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRa
te,true);
  fOutputs.AddOutputMetric(@CVD_L,@DummyProb,nil,
         'CVD Annual
Cases_LSLR',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
```

```
  fOutputs.AddOutputMetric(@CVDVal_L,@DummyProb,nil,
        'CVD
Annual_LSLR',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountR
ate,true);
  fOutputs.AddOutputMetric(@CVDVal,@DummyProb,nil,
        'CVD
Annual',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRate,t
rue);

  fOutputs.AddOutputMetric(@ADHD_C,@DummyProb,nil,
         'ADHD Annual
Cases_CCT',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@ADHDVal_C,@DummyProb,nil,
        'ADHD
Annual_CCT',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRa
te,true);
  fOutputs.AddOutputMetric(@ADHD_L,@DummyProb,nil,
         'ADHD Annual
Cases_LSLR',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@ADHDVal_L,@DummyProb,nil,
        'ADHD
Annual_LSLR',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountR
ate,true);
  fOutputs.AddOutputMetric(@ADHDVal,@DummyProb,nil,
        'ADHD
Annual',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRate,t
rue);

  fOutputs.AddOutputMetric(@LBW_C,@DummyProb,nil,
         'LBW Annual
Cases_CCT',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@LBWVal_C,@DummyProb,nil,
        'LBW
Annual_CCT',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRa
te,true);
  fOutputs.AddOutputMetric(@LBW_L,@DummyProb,nil,
         'LBW Annual
Cases_LSLR',mtBenefitCases,False,False,False,fConfig.OptionName,0,true);
  fOutputs.AddOutputMetric(@LBWVal_L,@DummyProb,nil,
        'LBW
Annual_LSLR',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountR
ate,true);
  fOutputs.AddOutputMetric(@LBWVal,@DummyProb,nil,
        'LBW
Annual',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRate,t
rue);

  fOutputs.AddOutputMetric(@TotBen,@DummyProb,nil,
```

```
        'Total Annual
Benefits',mtBenefitDollars,False,False,False,fConfig.OptionName,fConfig.DiscountRate
,true);
end;

destructor TBenefitsCollector.Destroy;
begin
  if fConfig.BenByYear then begin
    if assigned(BenefitsBaseline) then BenefitsBaseline.fBBYY.SaveResults;
    if assigned(BenefitsOption) then BenefitsOption.fBBYY.SaveResults;
  end;
  if assigned(BenefitsBaseline) then BenefitsBaseline.Free;
  if assigned(BenefitsOption) then BenefitsOption.Free;

  inherited;
end;

procedure TBenefitsCollector.GenerateBenefits(aCosts: TLCRCosts; ProxySystem :
boolean);
var i,f,t,y: integer;
    SmallSystem : boolean;
begin
  SmallSystem:=False;
  if (aCosts.CostingData.SystemType = 2) or  //ntnc = 2
     (aCosts.CostingData.Population <= fConfig.SmallProxyPop) then
     SmallSystem:=True;

  if fConfig.RunDifference then
  begin

BenefitsBaseline.CalcBenefitsNew(aCosts.BaseCostSteps.GMoveBinMicro,1,DoDebugOUt,
aCosts,
                                   aCosts.BaseCostSteps.pws90pctCCT_yr,
                                   aCosts.BaseCostSteps.pws90pctLSL_yr,
                                   -100000, -100000, SmallSystem,ProxySystem );
    BenefitsOption.CalcBenefitsNew(aCosts.ScenCostSteps.GMoveBinMicro,1,DoDebugOUt,
aCosts,
                                  aCosts.ScenCostSteps.pws90pctCCT_yr,
                                  aCosts.ScenCostSteps.pws90pctLSL_yr,
                                  bp1, bp2, SmallSystem,ProxySystem);

    for i:= low(FinalBins) to high(FinalBins) do begin
      FinalBins[i] := BenefitsOption.FinalBins[i] - BenefitsBaseline.FinalBins[i];
      StartingBins[i] := aCosts.ScenCostSteps.GMoveBin[0,i] -
aCosts.BaseCostSteps.GMoveBin[0,i];
      EndingBins[i] :=
aCosts.ScenCostSteps.GMoveBin[high(aCosts.ScenCostSteps.GMoveBin),i] -
```

```
aCosts.BaseCostSteps.GMoveBin[high(aCosts.BaseCostSteps.GMoveBin),i];
      EndingBinsCheck[i]:=aCosts.ScenCostSteps.GMoveBin[0,i];
    end;

    for y:=low(aCosts.ScenCostSteps.GMoveBinMicro) to
high(aCosts.ScenCostSteps.GMoveBinMicro) do begin
     for f:=low(aCosts.ScenCostSteps.GMoveBinMicro[y]) to
high(aCosts.ScenCostSteps.GMoveBinMicro[y]) do begin
       for t:=low(aCosts.ScenCostSteps.GMoveBinMicro[y,f]) to
high(aCosts.ScenCostSteps.GMoveBinMicro[y,f]) do begin
         if y=low(aCosts.ScenCostSteps.GMoveBinMicro) then begin
            BinMovements[f,t]:=aCosts.ScenCostSteps.GMoveBinTotal[f,t] -
aCosts.BaseCostSteps.GMoveBinTotal[f,t];
         end;
         if aCosts.ScenCostSteps.GMoveBinMicro[y,f,t]<0.01 then continue;
         EndingBinsCheck[f] := EndingBinsCheck[f] -
aCosts.ScenCostSteps.GMoveBinMicro[y,f,t];
         EndingBinsCheck[t] := EndingBinsCheck[t] +
aCosts.ScenCostSteps.GMoveBinMicro[y,f,t];
       end;
     end;
    end;

    BenPop := BenefitsOption.BenPop - BenefitsBaseline.BenPop;
    Ben7 := BenefitsOption.Ben7 - BenefitsBaseline.Ben7;
    Ben0 := BenefitsOption.Ben0 - BenefitsBaseline.Ben0;
    Ben11 := BenefitsOption.Ben11 - BenefitsBaseline.Ben11;
    BenA := BenefitsOption.BenA - BenefitsBaseline.BenA;

    IQLoss_C_BP1 := BenefitsOption.IQLoss_C_BP1 - BenefitsBaseline.IQLoss_C_BP1;
    IQLossVal_C_BP1 := BenefitsOption.IQLossVal_C_BP1 -
BenefitsBaseline.IQLossVal_C_BP1;
    IQLoss_L_Vol := BenefitsOption.IQLoss_L_Vol - BenefitsBaseline.IQLoss_L_Vol;
    IQLossVal_L_Vol := BenefitsOption.IQLossVal_L_Vol -
BenefitsBaseline.IQLossVal_L_Vol;
    IQLoss_C_BP2 := BenefitsOption.IQLoss_C_BP2 - BenefitsBaseline.IQLoss_C_BP2;
    IQLossVal_C_BP2 := BenefitsOption.IQLossVal_C_BP2 -
BenefitsBaseline.IQLossVal_C_BP2;
    IQLoss_L_Mand := BenefitsOption.IQLoss_L_Mand - BenefitsBaseline.IQLoss_L_Mand;
    IQLossVal_L_Mand := BenefitsOption.IQLossVal_L_Mand -
BenefitsBaseline.IQLossVal_L_Mand;
    IQLoss_L_Req := BenefitsOption.IQLoss_L_Req - BenefitsBaseline.IQLoss_L_Req;
    IQLossVal_L_Req := BenefitsOption.IQLossVal_L_Req -
BenefitsBaseline.IQLossVal_L_Req;
    IQLoss_POU := BenefitsOption.IQLoss_POU - BenefitsBaseline.IQLoss_POU;
    IQLossVal_POU := BenefitsOption.IQLossVal_POU - BenefitsBaseline.IQLossVal_POU;

    IQLossVal := BenefitsOption.IQLossVal - BenefitsBaseline.IQLossVal;
```

```
    CVD_C := BenefitsOption.CVD_C - BenefitsBaseline.CVD_C;
    CVDVal_C := BenefitsOption.CVDVal_C - BenefitsBaseline.CVDVal_C;
    CVD_L := BenefitsOption.CVD_L - BenefitsBaseline.CVD_L;
    CVDVal_L := BenefitsOption.CVDVal_L - BenefitsBaseline.CVDVal_L;
    CVDVal := BenefitsOption.CVDVal - BenefitsBaseline.CVDVal;

    ADHD_C := BenefitsOption.ADHD_C - BenefitsBaseline.ADHD_C;
    ADHDVal_C := BenefitsOption.ADHDVal_C - BenefitsBaseline.ADHDVal_C;
    ADHD_L := BenefitsOption.ADHD_L - BenefitsBaseline.ADHD_L;
    ADHDVal_L := BenefitsOption.ADHDVal_L - BenefitsBaseline.ADHDVal_L;
    ADHDVal := BenefitsOption.ADHDVal - BenefitsBaseline.ADHDVal;

    LBW_C := BenefitsOption.LBW_C - BenefitsBaseline.LBW_C;
    LBWVal_C := BenefitsOption.LBWVal_C - BenefitsBaseline.LBWVal_C;
    LBW_L := BenefitsOption.LBW_L - BenefitsBaseline.LBW_L;
    LBWVal_L := BenefitsOption.LBWVal_L - BenefitsBaseline.LBWVal_L;
    LBWVal := BenefitsOption.LBWVal - BenefitsBaseline.LBWVal;

    TotBen := BenefitsOption.TotBen - BenefitsBaseline.TotBen;
  end
  else if fConfig.RunBaselineOnly then
  begin

BenefitsBaseline.CalcBenefitsNew(aCosts.BaseCostSteps.GMoveBinMicro,1,DoDebugOUt,
aCosts,
                                    aCosts.BaseCostSteps.pws90pctCCT_yr,
                                    aCosts.BaseCostSteps.pws90pctLSL_yr,
                                    -10000, -100000, SmallSystem,ProxySystem);

    for i:= low(FinalBins) to high(FinalBins) do begin
      FinalBins[i] := BenefitsBaseline.FinalBins[i];
      StartingBins[i] := aCosts.BaseCostSteps.GMoveBin[0,i];
      EndingBins[i] :=
aCosts.BaseCostSteps.GMoveBin[high(aCosts.BaseCostSteps.GMoveBin),i];
      EndingBinsCheck[i]:=StartingBins[i];
    end;

    for y:=low(aCosts.BaseCostSteps.GMoveBinMicro) to
high(aCosts.BaseCostSteps.GMoveBinMicro) do begin
     for f:=low(aCosts.BaseCostSteps.GMoveBinMicro[y]) to
high(aCosts.BaseCostSteps.GMoveBinMicro[y]) do begin
       for t:=low(aCosts.BaseCostSteps.GMoveBinMicro[y,f]) to
high(aCosts.BaseCostSteps.GMoveBinMicro[y,f]) do begin
         if y=low(aCosts.BaseCostSteps.GMoveBinMicro) then begin
            BinMovements[f,t]:=aCosts.BaseCostSteps.GMoveBinTotal[f,t];
         end;
         if aCosts.BaseCostSteps.GMoveBinMicro[y,f,t]<0.01 then continue;
```

```
        EndingBinsCheck[f] := EndingBinsCheck[f] -
aCosts.BaseCostSteps.GMoveBinMicro[y,f,t];
        EndingBinsCheck[t] := EndingBinsCheck[t] +
aCosts.BaseCostSteps.GMoveBinMicro[y,f,t];
      end;
    end;
   end;

   BenPop := BenefitsBaseline.BenPop;
   Ben7 := BenefitsBaseline.Ben7;
   Ben0 := BenefitsBaseline.Ben0;
   Ben11 := BenefitsBaseline.Ben11;
   BenA := BenefitsBaseline.BenA;

   IQLoss_C_BP1 := BenefitsBaseline.IQLoss_C_BP1;
   IQLoss_C_BP2 := BenefitsBaseline.IQLoss_C_BP2;
   IQLossVal_C_BP1 := BenefitsBaseline.IQLossVal_C_BP1;
   IQLossVal_C_BP2 := BenefitsBaseline.IQLossVal_C_BP2;
   IQLoss_L_Vol := BenefitsBaseline.IQLoss_L_Vol;
   IQLoss_L_Mand := BenefitsBaseline.IQLoss_L_Mand;
   IQLoss_L_Req := BenefitsBaseline.IQLoss_L_Req;
   IQLossVal_L_Vol := BenefitsBaseline.IQLossVal_L_Vol;
   IQLossVal_L_Mand := BenefitsBaseline.IQLossVal_L_Mand;
   IQLossVal_L_Req := BenefitsBaseline.IQLossVal_L_Req;
   IQLossVal := BenefitsBaseline.IQLossVal;
   IQLoss_POU := BenefitsBaseline.IQLoss_POU;
   IQLossVal_POU := BenefitsBaseline.IQLossVal_POU;

   CVD_C := BenefitsBaseline.CVD_C;
   CVDVal_C := BenefitsBaseline.CVDVal_C;
   CVD_L := BenefitsBaseline.CVD_L;
   CVDVal_L := BenefitsBaseline.CVDVal_L;
   CVDVal := BenefitsBaseline.CVDVal;

   ADHD_C := BenefitsBaseline.ADHD_C;
   ADHDVal_C := BenefitsBaseline.ADHDVal_C;
   ADHD_L := BenefitsBaseline.ADHD_L;
   ADHDVal_L := BenefitsBaseline.ADHDVal_L;
   ADHDVal := BenefitsBaseline.ADHDVal;

   LBW_C := BenefitsBaseline.LBW_C;
   LBWVal_C := BenefitsBaseline.LBWVal_C;
   LBW_L := BenefitsBaseline.LBW_L;
   LBWVal_L := BenefitsBaseline.LBWVal_L;
   LBWVal := BenefitsBaseline.LBWVal;

   TotBen := BenefitsBaseline.TotBen;
  end
```

```
  else
  begin
    BenefitsOption.CalcBenefitsNew(aCosts.ScenCostSteps.GMoveBinMicro,1,DoDebugOUt,
aCosts,
                                   aCosts.ScenCostSteps.pws90pctCCT_yr,
                                   aCosts.ScenCostSteps.pws90pctLSL_yr,
                                   bp1, bp2, SmallSystem, ProxySystem);

    for i:= low(FinalBins) to high(FinalBins) do begin
      FinalBins[i] := BenefitsOption.FinalBins[i];
      StartingBins[i] := aCosts.ScenCostSteps.GMoveBin[0,i];
      EndingBins[i] :=
aCosts.ScenCostSteps.GMoveBin[high(aCosts.ScenCostSteps.GMoveBin),i];
      EndingBinsCheck[i]:=StartingBins[i];
    end;

    for y:=low(aCosts.ScenCostSteps.GMoveBinMicro) to
high(aCosts.ScenCostSteps.GMoveBinMicro) do begin
     for f:=low(aCosts.ScenCostSteps.GMoveBinMicro[y]) to
high(aCosts.ScenCostSteps.GMoveBinMicro[y]) do begin
       for t:=low(aCosts.ScenCostSteps.GMoveBinMicro[y,f]) to
high(aCosts.ScenCostSteps.GMoveBinMicro[y,f]) do begin
         if y=low(aCosts.ScenCostSteps.GMoveBinMicro) then begin
            BinMovements[f,t]:=aCosts.ScenCostSteps.GMoveBinTotal[f,t];
         end;
         if aCosts.ScenCostSteps.GMoveBinMicro[y,f,t]<0.01 then continue;
         EndingBinsCheck[f] := EndingBinsCheck[f] -
aCosts.ScenCostSteps.GMoveBinMicro[y,f,t];
         EndingBinsCheck[t] := EndingBinsCheck[t] +
aCosts.ScenCostSteps.GMoveBinMicro[y,f,t];
       end;
     end;
    end;

    BenPop := BenefitsOption.BenPop;
    Ben7 := BenefitsOption.Ben7;
    Ben0 := BenefitsOption.Ben0;
    Ben11 := BenefitsOption.Ben11;
    BenA := BenefitsOption.BenA;

    IQLoss_C_BP1 := BenefitsOption.IQLoss_C_BP1;
    IQLoss_C_BP2 := BenefitsOption.IQLoss_C_BP2;
    IQLossVal_C_BP1 := BenefitsOption.IQLossVal_C_BP1;
    IQLossVal_C_BP2 := BenefitsOption.IQLossVal_C_BP2;
    IQLoss_L_Vol := BenefitsOption.IQLoss_L_Vol;
    IQLoss_L_Mand := BenefitsOption.IQLoss_L_Mand;
    IQLoss_L_Req := BenefitsOption.IQLoss_L_Req;
    IQLossVal_L_Vol := BenefitsOption.IQLossVal_L_Vol;
```

```
    IQLossVal_L_Mand := BenefitsOption.IQLossVal_L_Mand;
    IQLossVal_L_Req := BenefitsOption.IQLossVal_L_Req;
    IQLossVal := BenefitsOption.IQLossVal;
    IQLoss_POU := BenefitsOption.IQLoss_POU;
    IQLossVal_POU := BenefitsOption.IQLossVal_POU;

    CVD_C := BenefitsOption.CVD_C;
    CVDVal_C := BenefitsOption.CVDVal_C;
    CVD_L := BenefitsOption.CVD_L;
    CVDVal_L := BenefitsOption.CVDVal_L;
    CVDVal := BenefitsOption.CVDVal;

    ADHD_C := BenefitsOption.ADHD_C;
    ADHDVal_C := BenefitsOption.ADHDVal_C;
    ADHD_L := BenefitsOption.ADHD_L;
    ADHDVal_L := BenefitsOption.ADHDVal_L;
    ADHDVal := BenefitsOption.ADHDVal;

    LBW_C := BenefitsOption.LBW_C;
    LBWVal_C := BenefitsOption.LBWVal_C;
    LBW_L := BenefitsOption.LBW_L;
    LBWVal_L := BenefitsOption.LBWVal_L;
    LBWVal := BenefitsOption.LBWVal;

    TotBen := BenefitsOption.TotBen;
  end;
end;

{ TLCRBenByYear }

procedure TLCRBenByYear.AddBen(aName: string);
begin
  fBenYears.Add(aName,TLCRBenYears.Create);
end;

constructor TLCRBenByYear.create(aConfig: TLCRConfig; aName : string);
begin
  inherited create;
  fBenYears := TObjectDictionary<string,TLCRBenYears>.create([doOwnsValues]);
  fTotYears:=aConfig.YearsOfOutput;
  Active:=aConfig.BenByYear;
  fOutName :=  USerPath+aConfig.RunName+'_'+aName+'_BenByYear.csv';
end;

destructor TLCRBenByYear.Destroy;
begin
  fBenYears.Free;
  inherited;
```

```
end;

procedure TLCRBenByYear.SaveResults;
var M : array of array of double;
    S,ss,HL : string;
    i,j : integer;
    T : TLCRBenYears;
    TOut : TBufferedFileStream;
    DoIt : boolean;
begin
  if not Active then exit;
  setlength(M,fBenYears.Count,150);
  HL:='Year';
  i:=0;
  for s in fBenYears.Keys do begin
    HL:=HL+','+S;
    T:=fBenYears.Items[S];
    for j:=1 to 149 do begin
      M[i,j-1]:=T.Year[j];
    end;
    inc(i);
  end;
  TOut := TBufferedFileStream.Create(fOutName,fmCreate);
  HL:=HL+#13#10;
  TOut.WriteBuffer(HL[1], Length(HL)*SizeOf(Char));
  for j:=1 to 150 do begin
    ss:=j.ToString;
    DoIt := False;
    for i:= 0 to fBenYears.Count - 1 do begin
      ss:=ss+','+M[i,j-1].ToString;
      if M[i,j-1]  <> 0 then Doit := True;
    end;
    ss:=ss+#13#10;
    if DoIt then
      TOut.WriteBuffer(ss[1], Length(ss)*SizeOf(Char));
  end;
  TOut.Free;
end;

procedure TLCRBenByYear.UpdateBen(Yr: integer; const aName: string; aValue: double);
var T : TLCRBenYears;
    y : integer;
begin
  if not Active then exit;
  // IncomeDeltaAge is a total over 10 years....
  if fBenYears.TryGetValue(aName,T) then begin
     for y:=6 to high(IncomeDeltaAge) do begin
       T.Year[Yr + (y - 6)] := T.Year[Yr + (y - 6)] + aValue * ( IncomeDeltaAge[y] /
```

```
10);
     end;
  end;
end;

initialization


PBF:=
'0,3.612,2.349,0.968,2.566,2.566,2.566,1.724,1.724,1.724,0.968,0.968,0.968,1.853,1.3
59,0.968,0.968'+#13#10+
'1,2.468,1.826,1.141,1.926,1.926,1.926,1.517,1.517,1.517,1.141,1.141,1.141,1.573,1.3
34,1.141,1.141'+#13#10+
'2,2.645,1.881,1.177,2.047,2.047,2.047,1.570,1.570,1.570,1.177,1.177,1.177,1.644,1.3
56,1.177,1.177'+#13#10+
'3,2.473,1.812,1.155,1.952,1.952,1.952,1.537,1.537,1.537,1.155,1.155,1.155,1.604,1.3
37,1.155,1.155'+#13#10+
'4,2.480,1.813,1.138,1.937,1.937,1.937,1.514,1.514,1.514,1.138,1.138,1.138,1.574,1.3
19,1.138,1.138'+#13#10+
'5,2.656,1.876,1.188,2.032,2.032,2.032,1.577,1.577,1.577,1.188,1.188,1.188,1.632,1.3
71,1.188,1.188'+#13#10+
'6,2.340,1.650,0.982,1.756,1.756,1.756,1.369,1.369,1.369,0.982,0.982,0.982,1.432,1.1
94,0.982,0.982'+#13#10+
'7,2.586,1.836,1.065,1.955,1.955,1.955,1.502,1.502,1.502,1.065,1.065,1.065,1.559,1.2
84,1.065,1.065'+#13#10+

'20,1.30548208,0.88611232,0.55575112,0.95088136,0.95088136,0.95088136,0.72512368,0.7
2512368,0.72512368,0.55575112,0.55575112,0.55575112,0.75792088,0.63472504,0.55575112
,0.55575112'+#13#10+
'21,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'22,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'23,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'24,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'25,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'26,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'27,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
```

```
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'28,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'29,1.60166524,1.02416296,0.56923186,1.11335458,1.11335458,1.11335458,0.80247004,0.8
0247004,0.80247004,0.56923186,0.56923186,0.56923186,0.84763414,0.67798462,0.56923186
,0.56923186'+#13#10+
'30,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'31,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'32,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'33,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'34,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'35,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'36,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'37,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'38,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'39,1.69166524,1.11416296,0.65923186,1.20335458,1.20335458,1.20335458,0.89247004,0.8
9247004,0.89247004,0.65923186,0.65923186,0.65923186,0.93763414,0.76798462,0.65923186
,0.65923186'+#13#10+
'40,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'41,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'42,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'43,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
```

```
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'44,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'45,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'46,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'47,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'48,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'49,1.89166524,1.31416296,0.85923186,1.40335458,1.40335458,1.40335458,1.09247004,1.0
9247004,1.09247004,0.85923186,0.85923186,0.85923186,1.13763414,0.96798462,0.85923186
,0.85923186'+#13#10+
'50,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'51,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'52,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'53,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'54,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'55,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'56,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'57,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'58,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'59,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
```

```
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'60,2.30166524,1.72416296,1.26923186,1.81335458,1.81335458,1.81335458,1.50247004,1.5
0247004,1.50247004,1.26923186,1.26923186,1.26923186,1.54763414,1.37798462,1.26923186
,1.26923186'+#13#10+
'61,2.30166524,1.72416296,1.26923186,1.81335458,1.81335458,1.81335458,1.50247004,1.5
0247004,1.50247004,1.26923186,1.26923186,1.26923186,1.54763414,1.37798462,1.26923186
,1.26923186'+#13#10+
'62,2.30166524,1.72416296,1.26923186,1.81335458,1.81335458,1.81335458,1.50247004,1.5
0247004,1.50247004,1.26923186,1.26923186,1.26923186,1.54763414,1.37798462,1.26923186
,1.26923186'+#13#10+
'63,2.30166524,1.72416296,1.26923186,1.81335458,1.81335458,1.81335458,1.50247004,1.5
0247004,1.50247004,1.26923186,1.26923186,1.26923186,1.54763414,1.37798462,1.26923186
,1.26923186'+#13#10+
'64,2.30166524,1.72416296,1.26923186,1.81335458,1.81335458,1.81335458,1.50247004,1.5
0247004,1.50247004,1.26923186,1.26923186,1.26923186,1.54763414,1.37798462,1.26923186
,1.26923186'+#13#10+
'65,2.40690248,1.77321392,1.27402172,1.87108316,1.87108316,1.87108316,1.52995208,1.5
2995208,1.52995208,1.27402172,1.27402172,1.27402172,1.57951028,1.39335524,1.27402172
,1.27402172'+#13#10+
'66,2.40690248,1.77321392,1.27402172,1.87108316,1.87108316,1.87108316,1.52995208,1.5
2995208,1.52995208,1.27402172,1.27402172,1.27402172,1.57951028,1.39335524,1.27402172
,1.27402172'+#13#10+
'67,2.40690248,1.77321392,1.27402172,1.87108316,1.87108316,1.87108316,1.52995208,1.5
2995208,1.52995208,1.27402172,1.27402172,1.27402172,1.57951028,1.39335524,1.27402172
,1.27402172'+#13#10+
'68,2.40690248,1.77321392,1.27402172,1.87108316,1.87108316,1.87108316,1.52995208,1.5
2995208,1.52995208,1.27402172,1.27402172,1.27402172,1.57951028,1.39335524,1.27402172
,1.27402172'+#13#10+
'69,2.40690248,1.77321392,1.27402172,1.87108316,1.87108316,1.87108316,1.52995208,1.5
2995208,1.52995208,1.27402172,1.27402172,1.27402172,1.57951028,1.39335524,1.27402172
,1.27402172'+#13#10+
'70,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'71,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'72,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'73,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'74,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'75,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
```

```
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'76,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'77,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'78,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'79,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'+#13#10+
'80,2.51690248,1.88321392,1.38402172,1.98108316,1.98108316,1.98108316,1.63995208,1.6
3995208,1.63995208,1.38402172,1.38402172,1.38402172,1.68951028,1.50335524,1.38402172
,1.38402172'
;

PBM:=
'0,3.612,2.349,0.968,2.566,2.566,2.566,1.724,1.724,1.724,0.968,0.968,0.968,1.853,1.3
59,0.968,0.968'+#13#10+
'1,2.468,1.826,1.141,1.926,1.926,1.926,1.517,1.517,1.517,1.141,1.141,1.141,1.573,1.3
34,1.141,1.141'+#13#10+
'2,2.645,1.881,1.177,2.047,2.047,2.047,1.570,1.570,1.570,1.177,1.177,1.177,1.644,1.3
56,1.177,1.177'+#13#10+
'3,2.473,1.812,1.155,1.952,1.952,1.952,1.537,1.537,1.537,1.155,1.155,1.155,1.604,1.3
37,1.155,1.155'+#13#10+
'4,2.480,1.813,1.138,1.937,1.937,1.937,1.514,1.514,1.514,1.138,1.138,1.138,1.574,1.3
19,1.138,1.138'+#13#10+
'5,2.656,1.876,1.188,2.032,2.032,2.032,1.577,1.577,1.577,1.188,1.188,1.188,1.632,1.3
71,1.188,1.188'+#13#10+
'6,2.340,1.650,0.982,1.756,1.756,1.756,1.369,1.369,1.369,0.982,0.982,0.982,1.432,1.1
94,0.982,0.982'+#13#10+
'7,2.586,1.836,1.065,1.955,1.955,1.955,1.502,1.502,1.502,1.065,1.065,1.065,1.559,1.2
84,1.065,1.065'+#13#10+

'20,1.60548208,1.18611232,0.85575112,1.25088136,1.25088136,1.25088136,1.02512368,1.0
2512368,1.02512368,0.85575112,0.85575112,0.85575112,1.05792088,0.93472504,0.85575112
,0.85575112'+#13#10+
'21,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'22,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'23,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
```

```
,0.86923186'+#13#10+
'24,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'25,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'26,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'27,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'28,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'29,1.90166524,1.32416296,0.86923186,1.41335458,1.41335458,1.41335458,1.10247004,1.1
0247004,1.10247004,0.86923186,0.86923186,0.86923186,1.14763414,0.97798462,0.86923186
,0.86923186'+#13#10+
'30,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'31,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'32,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'33,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'34,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'35,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'36,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'37,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'38,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
,0.98923186'+#13#10+
'39,2.02166524,1.44416296,0.98923186,1.53335458,1.53335458,1.53335458,1.22247004,1.2
2247004,1.22247004,0.98923186,0.98923186,0.98923186,1.26763414,1.09798462,0.98923186
```

```
,0.98923186'+#13#10+
'40,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'41,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'42,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'43,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'44,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'45,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'46,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'47,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'48,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'49,2.22166524,1.64416296,1.18923186,1.73335458,1.73335458,1.73335458,1.42247004,1.4
2247004,1.42247004,1.18923186,1.18923186,1.18923186,1.46763414,1.29798462,1.18923186
,1.18923186'+#13#10+
'50,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'51,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'52,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'53,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'54,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'55,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
```

```
,1.38923186'+#13#10+
'56,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'57,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'58,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'59,2.42166524,1.84416296,1.38923186,1.93335458,1.93335458,1.93335458,1.62247004,1.6
2247004,1.62247004,1.38923186,1.38923186,1.38923186,1.66763414,1.49798462,1.38923186
,1.38923186'+#13#10+
'60,2.58166524,2.00416296,1.54923186,2.09335458,2.09335458,2.09335458,1.78247004,1.7
8247004,1.78247004,1.54923186,1.54923186,1.54923186,1.82763414,1.65798462,1.54923186
,1.54923186'+#13#10+
'61,2.58166524,2.00416296,1.54923186,2.09335458,2.09335458,2.09335458,1.78247004,1.7
8247004,1.78247004,1.54923186,1.54923186,1.54923186,1.82763414,1.65798462,1.54923186
,1.54923186'+#13#10+
'62,2.58166524,2.00416296,1.54923186,2.09335458,2.09335458,2.09335458,1.78247004,1.7
8247004,1.78247004,1.54923186,1.54923186,1.54923186,1.82763414,1.65798462,1.54923186
,1.54923186'+#13#10+
'63,2.58166524,2.00416296,1.54923186,2.09335458,2.09335458,2.09335458,1.78247004,1.7
8247004,1.78247004,1.54923186,1.54923186,1.54923186,1.82763414,1.65798462,1.54923186
,1.54923186'+#13#10+
'64,2.58166524,2.00416296,1.54923186,2.09335458,2.09335458,2.09335458,1.78247004,1.7
8247004,1.78247004,1.54923186,1.54923186,1.54923186,1.82763414,1.65798462,1.54923186
,1.54923186'+#13#10+
'65,2.68690248,2.05321392,1.55402172,2.15108316,2.15108316,2.15108316,1.80995208,1.8
0995208,1.80995208,1.55402172,1.55402172,1.55402172,1.85951028,1.67335524,1.55402172
,1.55402172'+#13#10+
'66,2.68690248,2.05321392,1.55402172,2.15108316,2.15108316,2.15108316,1.80995208,1.8
0995208,1.80995208,1.55402172,1.55402172,1.55402172,1.85951028,1.67335524,1.55402172
,1.55402172'+#13#10+
'67,2.68690248,2.05321392,1.55402172,2.15108316,2.15108316,2.15108316,1.80995208,1.8
0995208,1.80995208,1.55402172,1.55402172,1.55402172,1.85951028,1.67335524,1.55402172
,1.55402172'+#13#10+
'68,2.68690248,2.05321392,1.55402172,2.15108316,2.15108316,2.15108316,1.80995208,1.8
0995208,1.80995208,1.55402172,1.55402172,1.55402172,1.85951028,1.67335524,1.55402172
,1.55402172'+#13#10+
'69,2.68690248,2.05321392,1.55402172,2.15108316,2.15108316,2.15108316,1.80995208,1.8
0995208,1.80995208,1.55402172,1.55402172,1.55402172,1.85951028,1.67335524,1.55402172
,1.55402172'+#13#10+
'70,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'71,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
```

```
,1.75402172'+#13#10+
'72,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'73,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'74,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'75,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'76,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'77,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'78,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'79,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'+#13#10+
'80,2.88690248,2.25321392,1.75402172,2.35108316,2.35108316,2.35108316,2.00995208,2.0
0995208,2.00995208,1.75402172,1.75402172,1.75402172,2.05951028,1.87335524,1.75402172
,1.75402172'
;

PBSens1 :=
'0,3.612,2.349,1.152,2.566,2.566,2.566,1.724,1.724,1.724,1.152,1.152,1.152,1.853,1.3
59,0.968,0.968'+#13#10+
'1,2.468,1.826,1.228,1.926,1.926,1.926,1.517,1.517,1.517,1.228,1.228,1.228,1.573,1.3
34,1.141,1.141'+#13#10+
'2,2.645,1.881,1.250,2.047,2.047,2.047,1.570,1.570,1.570,1.250,1.250,1.250,1.644,1.3
56,1.177,1.177'+#13#10+
'3,2.473,1.812,1.242,1.952,1.952,1.952,1.537,1.537,1.537,1.242,1.242,1.242,1.604,1.3
37,1.155,1.155'+#13#10+
'4,2.480,1.813,1.219,1.937,1.937,1.937,1.514,1.514,1.514,1.219,1.219,1.219,1.574,1.3
19,1.138,1.138'+#13#10+
'5,2.656,1.876,1.254,2.032,2.032,2.032,1.577,1.577,1.577,1.254,1.254,1.254,1.632,1.3
71,1.188,1.188'+#13#10+
'6,2.340,1.650,1.071,1.756,1.756,1.756,1.369,1.369,1.369,1.071,1.071,1.071,1.432,1.1
94,0.982,0.982'
;

PBSens2 :=
```

```
'0,3.612,2.349,1.259,2.566,2.566,2.566,1.724,1.724,1.724,1.259,1.259,1.259,1.853,1.3
59,0.968,0.968'+#13#10+
'1,2.468,1.826,1.281,1.926,1.926,1.926,1.517,1.517,1.517,1.281,1.281,1.281,1.573,1.3
34,1.141,1.141'+#13#10+
'2,2.645,1.881,1.311,2.047,2.047,2.047,1.570,1.570,1.570,1.311,1.311,1.311,1.644,1.3
56,1.177,1.177'+#13#10+
'3,2.473,1.812,1.296,1.952,1.952,1.952,1.537,1.537,1.537,1.296,1.296,1.296,1.604,1.3
37,1.155,1.155'+#13#10+
'4,2.480,1.813,1.274,1.937,1.937,1.937,1.514,1.514,1.514,1.274,1.274,1.274,1.574,1.3
19,1.138,1.138'+#13#10+
'5,2.656,1.876,1.314,2.032,2.032,2.032,1.577,1.577,1.577,1.314,1.314,1.314,1.632,1.3
71,1.188,1.188'+#13#10+
'6,2.340,1.650,1.126,1.756,1.756,1.756,1.369,1.369,1.369,1.126,1.126,1.126,1.432,1.1
94,0.982,0.982';
;

PBSens3 :=
'0,3.156,2.119,0.980,2.234,2.234,2.234,1.608,1.608,1.608,0.980,0.980,0.980,1.675,1.2
61,0.980,0.980'+#13#10+
'1,2.228,1.679,1.141,1.778,1.778,1.778,1.441,1.441,1.441,1.141,1.141,1.141,1.485,1.2
68,1.141,1.141'+#13#10+
'2,2.350,1.749,1.160,1.849,1.849,1.849,1.479,1.479,1.479,1.160,1.160,1.160,1.523,1.3
26,1.160,1.160'+#13#10+
'3,2.254,1.690,1.142,1.777,1.777,1.777,1.438,1.438,1.438,1.142,1.142,1.142,1.494,1.2
91,1.142,1.142'+#13#10+
'4,2.270,1.711,1.156,1.772,1.772,1.772,1.446,1.446,1.446,1.156,1.156,1.156,1.485,1.2
83,1.156,1.156'+#13#10+
'5,2.341,1.774,1.195,1.870,1.870,1.870,1.512,1.512,1.512,1.195,1.195,1.195,1.541,1.3
15,1.195,1.195'+#13#10+
'6,2.097,1.527,1.001,1.602,1.602,1.602,1.276,1.276,1.276,1.001,1.001,1.001,1.317,1.1
25,1.001,1.001';
;


end.
```