Information on the Main Computer Codes Used in the Dose Assessment

Contents

A9.1. INTRODUCTION

The estimation of the thyroid doses received by the American people from ^{131}I in fallout from Nevada bomb tests is carried out in three steps:

1. Assessment of the extent to which ^{131}I was deposited per unit area of ground. (Step 1)

2. Estimation of the concentrations of ^{131}I in several categories of cows' milk (i.e., fresh cows' milk, milk consumed on the farm, milk sold retail in the same county, milk that originated in another county within the same milk marketing region, volume-weighted mixed milk, and milk obtained from a backyard cow) and in goats' milk, cottage cheese, eggs, leafy vegetables, and ground-level air. (Step 2)

3. Assessment of average thyroid doses for various population groups (i.e., those persons who consumed average diets, the "high-exposure" groups, the "low-exposure" groups, persons who drank milk from backyard cows, and infants who were fed mother's milk) and of the per capita doses. (Step 3)

Those three steps are illustrated in *Figure A9.1.*

A9.2. STEP 1. ESTIMATION OF ^{131}I RADIOACTIVITY DEPOSITED PER UNIT AREA OF GROUND

A schematic representation of the procedure used to derive the daily depositions of ^{131}I in the 3,094 counties and sub-counties of the contiguous United States is provided in *Figure A9.2.*

When gummed-film data were available from DOE/EML, which is the situation for most of the tests, the daily depositions of ^{131}I were estimated for the 3,071 counties of the U.S. together with the use of either the kriging or the AIPC method, and the precipitation data supplied by NOAA/ARL. The computer programs used with the kriging method were developed by EML staff[1]. The computer program implementing the AIPC method is called DEPINTER.FOR (**Attachment A9.1**).

When gummed-film data were not available (nine tests, **Section 3.3.2**), a meteorological model was applied to obtain estimates of daily depositions in 3,071 counties. The computer programs associated with the meteorological model were developed by NOAA staff[1].

Monitoring data from locations in counties and states near the NTS were supplied by DOE/NVO in the County Data Base (CDB) and in the Town Data Base (TDB). These data, which are available for almost all of the tests, are expressed in terms of exposure rates at H + 12 associated with estimated initial times of arrival, and were converted to depositions of ^{131}I per unit area of ground in those locales by the computer programs TDB1089.FOR (**Attachment A9.2**) and CDB290.FOR (**Attachment A9.3**).

[1] Programs developed by agencies other than NCI remain in the possession of those agencies.

Only the most complete and valid database and/or analytical method available for a geographical area was used as a basis for ^{131}I deposition estimates:

Town Data Base:	five counties in Nevada and Utah (**Section 3.2.1**).
County Data Base:	134 counties in Arizona, California, Colorado, Idaho, New Mexico, Nevada, Oregon, Utah, and Wyoming (**Section 3.2.1**).
Gummed-film Data:	the remaining 2,937 counties in the U.S. at the time.
Meteorological Model:	the remaining 2,937 counties in the U.S. at that time when gummed-film data were not available (nine tests, **Section 3.3.2**)

The daily deposition estimates of ^{131}I from the (a) Town Data Base, (b) County Data Base, (c) gummed-film data, to which either the kriging or AIPC method was applied, and (d) the meteorological model were combined using the program DEP(test name).FOR (**Attachment A9.4**). This program will provide the estimated daily depositions of ^{131}I per unit area of ground for any of the nuclear tests included in this analysis, using the most complete and valid of the several sources of deposition estimates. In this Appendix, the test Simon is used as an example,so that the program DEPSIMON.FOR is provided as **Attachment A9.4.**

A9.3. STEP 2. ESTIMATION OF THE CONCENTRATION OF ^{131}I IN FOODSTUFFS AND IN AIR

A schematic representation of the procedure used to derive the concentration of ^{131}I in foodstuffs and in air for the 3,094 counties and sub-counties of the contiguous United States from the daily depositions of ^{131}I per unit area of ground is provided as *Figure A9.3.* A computer program called CONCUST9.FOR (**Attachment A9.5**) calculates those concentrations for each day of fallout resulting from a particular nuclear test, and sums those concentrations for each day of fallout on the basis of:

(a) the calculated distance of each county centroid from the NTS,

(b) the precipitation for each day and for each county,

(c) the pasture intake by cows for each day and for each county, and

(d) the volumes of milk available for fluid use in each county and milk transferred into or out of the county.

The computer program CONCUST9.FOR includes:

- the distance from the NTS calculated by DISTANCE.FOR (**Attachment A9.6**),
- the precipitation data provided by NOAA/ARL,
- the pasture data calculated by NUPAST.FOR (**Attachment A9.7**) and NEWPASTREG.FOR (**Attachment A9.8**),
- the milk production, utilization, and distribution data calculated using MILLER.FOR (**Attachment A9.9**), NEWMILLERUS2.FOR (**Attachment A9.10**), and MILKDIST.FOR (**Attachment A9.11**).

A9.4. STEP 3. ESTIMATION OF AVERAGE THYROID DOSES

A schematic representation of the procedure used to derive estimates of average thyroid doses for various population groups in the 3,094 counties and sub-counties of the contiguous United States from the time-integrated concentrations of ^{131}I in cows' milk, other foodstuffs, and ground-level air is provided in *Figure A9.4*. The computer program GRPDOSE1.FOR (**Attachment A9.12**) estimates the average doses to various population groups using a consumption data file prepared by hand, while the program PERCAP1.FOR (**Attachment A9.13**) estimates the per capita and collective doses for the 3,094 counties and sub-counties of the contiguous U.S. using population data provided by the Environmental Protection Agency.

Figure A9.1. Schematic representation of the steps used to estimate thyroid doses.

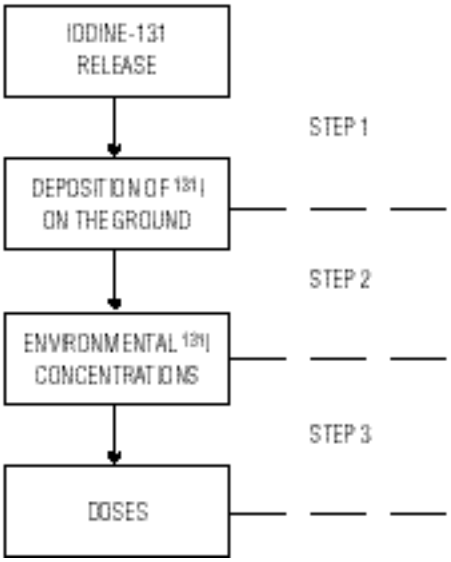


Figure A9.2. Schematic representation of the procedure used to estimate daily depositions of ^{131}I in each of the 3,094 counties and sub-counties of the contiguous U.S.

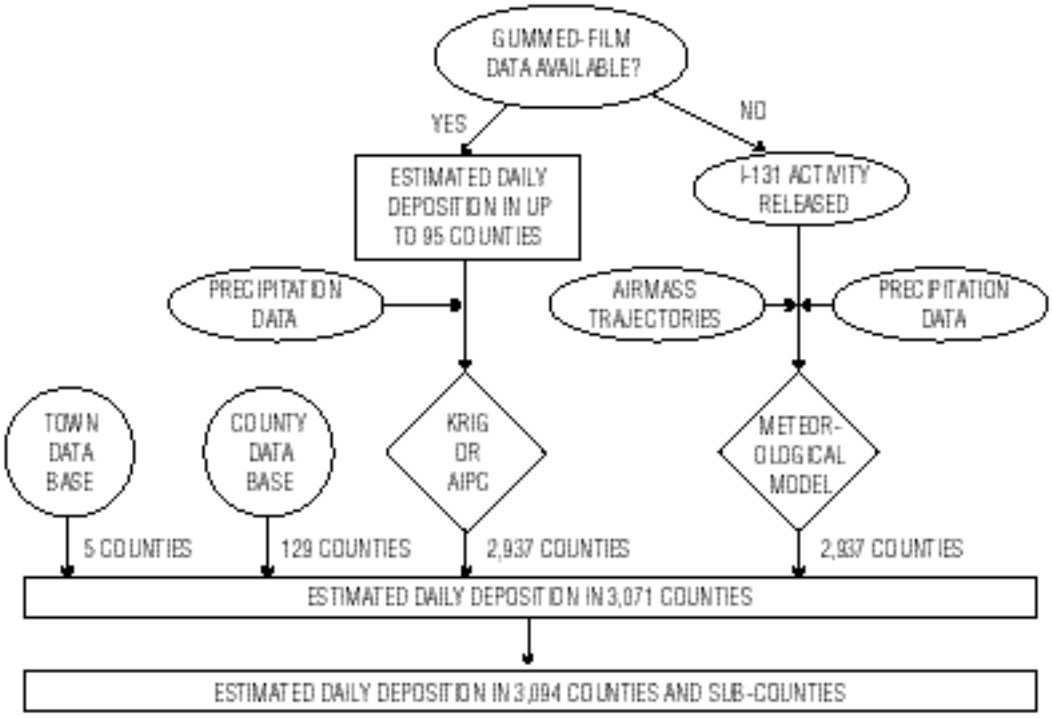


Figure A9.3. Schematic representation of the procedure used to estimate time-integrated concentrations of ^{131}I in the 3,094 counties and sub-counties of the contiguous U.S.

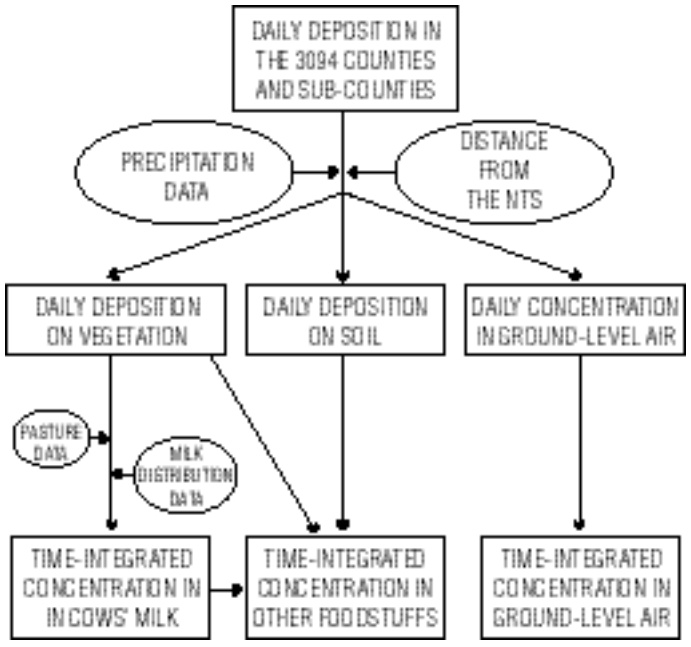
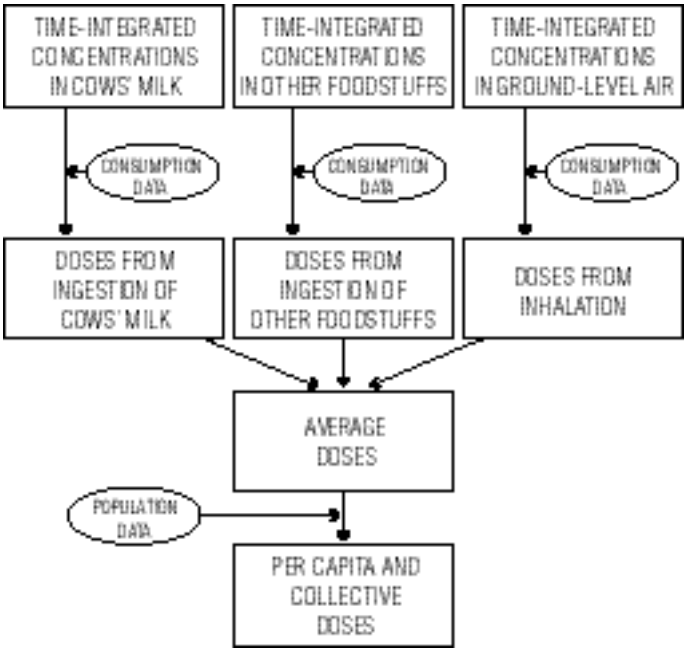


Figure A9.4. Schematic representation of the procedure used to calculate average ^{131}I thyroid doses for population groups in the 3,094 counties and sub-counties of the contiguous U.S.



ATTACHMENT A9.1: PROGRAM DEPINTER.FOR

```
      program depinter
c
c
c     interpolation by "hand": incorporates precip data, uncertainties,
c     and estimated depositions into a GF file.
c     Files: . (test)(date).HAND is the daily GF input file
c              copied from disk$bouville:[andre.maps](test)(date).HAND
c     . [dreicer.milk]BO2.DAT is the FIPS file with county
c       coordinates
c     . PRE(date)F.DAT is the daily precip file copied from
c         disk$bouville:[andre.precip.daily]
c     ***    . GFST.DAT is the GF station file
c     . (test)H(date).FIN is the daily "interpolated" output file
c
c
      dimension fps(130),prec(130),dp(130),cf(10),cgf(130),ef(10)
      dimension gflat(130),gflon(130)
      character*12 GFNAME,bjname
      character*8 TNAME
      integer fps,prec,dp,fips,dep
c
      open (unit = 1, file = 'gfx.dat', status = 'old')
      open (unit = 2, file = '[dreicer.milk]b01.tape', status = 'old')
      open (unit = 3, file = 'day.dat', status = 'old')
      open (unit = 4, file = 'gfst.dat', status = 'old')
      open (unit = 10, file = 'gfx.fin', status = 'new')
      open (unit = 11, file = 'check.dat', status = 'new')
c
      cf(1) = 1.
      cf(2) = 1.5
      cf(3) = 2.
      cf(4) = 2.
      cf(5) = 4.
      cf(6) = 6.
      cf(7) = 10.
      cf(8) = 10.
      cf(9) = 10.
c
      ef(1) = 20. / 20.
      ef(2) = 20. / 20.
      ef(3) = 20. / 30.
      ef(4) = 20. / 25.
      ef(5) = 20. / 15.
      ef(6) = 20. / 10.
      ef(7) = 20. / 7.
      ef(8) = 20. / 7.
      ef(9) = 20. / 7.
c
      read (1,100) tname,im,it,iy,im,id
      read (4,400)
      do 12 k = 1,130
      read (4,401,end=13) bjname,fps(k),gflat(k),gflon(k)
      read (1,101,end=13) gfname,dp(k),prec(k)
      np = prec(k)
c     dp(k) = dp(k) * ef(np)
```

```
            cgf(k) = cf(np)
            if (bjname.ne.gfname) write (11,110) k,bjname,gfname
12          continue
13          kmax = k - 1
c
            write (10,510) tname,im,it,iy
            n = 0
            iyes = 0
            ino = 0
1           CONTINUE
            read (2,200,end=20) nfips,ulat,ulon
            n = n + 1
            dref = 300.
            dep = 0
            kk = 130
            do 2 k = 1,kmax
            if (dp(k).lt.0) go to 2
            x = abs(gflat(k) - ulat)
            y = abs(gflon(k) - ulon)
            if (x.gt.10.) go to 2
            if (y.gt.10.) go to 2
            d = x*x + y*y
            if (d.ge.dref) go to 2
            dref = d
            kk = k
2           continue
c
c           write (11,512) kmax,n,kk,dp(kk),d
c
            mm = 1
            dpp = dp(kk)
            if (dref.eq.300.) go to 5
            mm = 2
            dpp = dp(kk)
            if (dp(kk).eq.0) go to 5
            udep = log(3.)
            if (nfips.eq.fps(kk)) udep = log(1.5)
3           read (3,300,end=4) im,id,iyear,fips,iprecip
            if (fips.ne.nfips) go to 3
            if (nfips.eq.fps(kk)) iprecip = prec(kk)
            dpp = dp(kk) * cf(iprecip) / cgf(kk)
            dep = dpp
            mm = 3
            if (dep.eq.0) go to 5
            iyes = iyes + 1
            write (10,511) nfips,tname,id,dep,udep,iprecip
            go to 1
4           iprecip = 1
            if (nfips.eq.fps(kk)) iprecip = prec(kk)
            dpp = dp(kk) * cf(iprecip) / cgf(kk)
            dep = dpp
            mm = 4
            if (dep.eq.0) go to 5
            iyes = iyes + 1
            write (10,511) nfips,tname,id,dep,udep,iprecip
            rewind 3
            go to 1
```

```
5         continue
          ino = ino + 1
          write (11,512) mm,n,nfips,kk,dpp,dref,iprecip
          go to 1
20        continue
          itot = iyes + ino
          write (11,513) iyes,ino,itot
513       format (3i5)
100       format (1x,a,i2,1x,i2,1x,i2,13x,i3,1x,i2)
101       format (1x,a,4x,i5,i7)
110       format (i5,a,2x,a)
200       format (i5,34x,2f8.3)
300       format (5x,i2,1x,i2,1x,i4,i8,i5)
400       format (//)
401       format (a,7x,i5,f7.2,f8.2)
510       format (/2x,'HAND :',a,'(',i2,'/',i2,'/',i2,')',1x,'I'/
     1'FIPS',7X,'SHOT',7X,'DAY',2X,'131',4X,'ERR',2X,'PRECIP'/)
511       format (i5,1x,a,i10,i8,f6.2,i5)
512       format (4i6,2f8.3,i5)
          stop
          end
```

ATTACHMENT A9.2 : PROGRAM TDB1089.FOR

```
          PROGRAM TDB1089
C
C       CALCULATES THE I-131 DEPOSITION DENSITIES NEAR NTS FROM THE
C       EXPOSURE RATES AND TOA GIVEN IN THE TOWN DATA BASE
C       ONE PROGRAM FOR EACH TEST : CHANGE THE NAME OF THE TEST
C       revised april 1989 : GSD on deposition assumed to be ge 1.4
c       REVISED 30 january 1990
C       OUTPUT FILE TDB(test).RES : results for the 173 TDB stations
C       OUTPUT FILE TDB(test).FIN : results for the 13 sub-counties
C       OUTPUT FILE TDB(test).CTY : results for the 5 counties (for map
c       purposes)
C
          DIMENSION IPS(60),IPSC(5),NUMC(5),AREAC(5),DEPC(5)
          DIMENSION NUM(60),RNAME(4),RI(4),AREA(60),DEPSC(20)
          DIMENSION IPS1(2020),DEP(2020),H12E(2020),DEPCE(5)
          DIMENSION DEPSCE(20),npb(4),x(2020),s(2020),xsc(20),CTYSC(13)
         dimension s2xsc(20),xc(5),s2xc(5),STC(5),CTYC(5),STSC(13)
         REAL LON,LAT,mu,muxsc,muxc
         CHARACTER*10 TNAME,TN,RNAME
         CHARACTER*16 TT
         CHARACTER*2 STC,STSC,SERIES
         CHARACTER*10 CTYC,CTYSC
          OPEN (UNIT=1,FILE='TDBTEST.DAT', STATUS='OLD')
          OPEN (UNIT=2,FILE='TDBNUM1089.CTY', STATUS='OLD')
          OPEN (UNIT=3,FILE='EXTEST.DAT',STATUS='OLD')
          OPEN (UNIT=4,FILE='TDBW.DAT',STATUS='OLD')
          OPEN (UNIT=14,FILE='TDBTEST.RES',STATUS='NEW')
          OPEN (UNIT=15,FILE='TDBTEST.FIN',STATUS='NEW')
          OPEN (UNIT=16,FILE='TDBTEST.CTY',STATUS='NEW')
          OPEN (UNIT=17,FILE='check.dat',STATUS='NEW')
         DO 1 J=1,13
```

```
1       READ (2,100) I,STSC(I),CTYSC(I),IPS(I),NUM(I),AREA(I)
100     FORMAT (I4,2X,A2,2X,A10,5X,I5,I7,F10.0)
        DO 2 J = 1,5
        READ (2,100) I,STC(I),CTYC(I),IPSC(I),NUMC(I),AREAC(I)
2       CONTINUE
        DO 10 I=1,2020
        DEP(I) = 0.
        H12E(I) = 1.
10      CONTINUE
c
c       read depositions (mmCi/m2 at H+12) corresponding to a total
c       exposure rate of 1 mR/h at H+12 for Sb-131,Te-131m, Te-131,
c        and I-131 and for this particular test
c
        READ (3,130) TNAME,ny,nm,nd,q,SERIES,NCODE
        ITEST = ny * 10000 + nm * 100 + nd
130     FORMAT (A,3I2,e10.2,4X,A2,1X,I2)
        DO 6 J=1,4
        READ (3,140) RNAME(J),RI(J)
6       CONTINUE
140     FORMAT (A,E10.3)
c
c       calculation of the "total" I-131 deposition density per
c       unit exposure rate at H + 12
c
        ri1 = ri(1)*23.3/(60.*8.04*24.)
        ri2 = ri(2)*30.0/(8.04*24.)
        ri3 = ri(3)*25.0/(60.*8.04*24.)
        RTI= RI1 + RI2 + RI3 + RI(4)
c
c       read Town Data Base data and calculate the median I-131 deposition
c       densities for all TDB stations
c
        DO 11 I = 1,300
        READ(1,110,END=3) ID,IPS1(ID),H12,H12E(ID),TOA
        RINT = RTI*EXP(-0.69315*(TOA-12.)/(8.04*24.))
        DEP(ID) = 1000.*H12*RINT
        if (h12e(id).lt.1.4) h12e(id) = 1.4
        mu = log (dep(id))
        sig = log (h12e(id))
        x(id) = exp (mu + (sig*sig/2.))
        s(id) = x(id) * sqrt(exp(sig*sig) - 1.)
11      CONTINUE
3       CONTINUE
c
c       prepare the first output file (depositions for all stations)
c
        write (14,219) SERIES,NCODE,tname,nm,nd,ny
219     format(1x,'Table ',A2,'/',I2,'/TDB. Estimates of median I-131',
     4' depositions ',
     1'per unit area of ground',/,12x,
     2' at the Town Data Base sites following the shot ',a,/,13x,
     3'detonated',1x,i2,'/',i2,'/19',i2,'.',/)
        write (14,218)
218     format (2x,'Test',6x,'Test',4x,'Site',1x,'State',2x,'County',
     14x,'Sub-',5x,'I-131',6x,'GSD',2x,'Deposition'/
     22x,'name',6x,'date',4x,'code',17x,'county',3x,
```

```
     3'deposition',9x,'weight'/
     411x,'(y/mo/d)',31x,'(nCi/m2)'/)
       write (15,228)
228    format (3x,'Test',7x,'Test',4x,'State',2x,'County',7x,'FIPS',
     17x,'I-131',4x,'GSD'/3x,'name',7x,'date',24x,'code',
     24x,'deposition'/
     312x,'(y/mo/d)',31x,'(nCi/m2)'/)
       Tl = 1
       npb(1) = 843
       npb(2) = 850
       npb(3) = 841
       gwr = 1.5
       sigwr = log(gwr)
       ND = 0
       DO 12 I = 1,13
       J = NUM(I)
       xsc(i) = 0.
       s2xsc(i) = 0.
       depsc(i) = 0.
       depsce(i) = 1.
       DO 13 K = 1,J
       READ (4,204) IPS2,ID,TT,WR
       ND = ND + 1
       n10 = ips2 / 10
       nsc = ips2 - 10 * n10
       WRITE (14,220) TNAME,ITEST,ND,STSC(I),CTYSC(I),nsc,DEP(ID),
     1H12E(ID),WR
       if (dep(id).eq.0.) go to 16
       xw = x(id) * wr
       xsc(i) = xsc(i) + xw
       sigx = sqrt(log(1. + ((s(id)/x(id))**2)))
       gxw = exp(sqrt(sigx*sigx + sigwr*sigwr))
       sigxw = log(gxw)
       sxw = xw * sqrt(exp(sigxw*sigxw) - 1.)
       s2xsc(i) = s2xsc(i) + (sxw*sxw)
16     continue
       if (id.ne.npb(l)) go to 13
       l = l + 1
       write (14,217) SERIES,NCODE
       write (14,218)
13     CONTINUE
c
c      print the second output file (depositions for all sub-counties)
c
       if (xsc(i).eq.0.) go to 17
       muxsc = log(xsc(i)/sqrt(1. + (s2xsc(i)/(xsc(i)**2))))
       DEPSC(I) = exp(muxsc)
       sigsc = sqrt(log(1. + (s2xsc(i)/(xsc(i)**2))))
       DEPSCE(I) = exp(sigsc)
17     continue
       WRITE (15,250) TNAME,ITEST,stsc(i),ctysc(i),IPS(I),DEPSC(I),
     1DEPSCE(I)
12     CONTINUE
c
c      prepare the third output file (depositions for all counties)
c
       write (16,228)
```

```
        N = 0
        xtot = 0.
        DO 14 I = 1,5
        J = NUMC(I)
        DEPC(I) = 0.
        DEPCE(I) = 1.
        xc(i) = 0.
        s2xc(i) = 0.
        muxc = 0.
        sigxc = 0.
        DO 15 K = 1,J
        N = N + 1
        XC(I) = XC(I) + (XSC(N)*AREA(N))
        xtot = xtot + (0.001 * XSC(N)*AREA(N))
        s2xc(i) = s2xc(i) + (s2xsc(n) * area(n) * area(n))
15      CONTINUE
        XC(I) = XC(I)/AREAC(I)
        s2xc(i) = s2xc(i) / (areac(i)**2)
        if (xc(i).eq.0.) go to 18
        muxc = log(xc(i)/sqrt(1. + (s2xc(i)/(xc(i)**2))))
        DEPC(I) = exp(muxc)
        sigxc = sqrt(log(1. + (s2xc(i)/(xc(i)**2))))
        DEPCE(I) = exp(sigxc)
18      continue
c       write (16,251) xc(i),s2xc(i),muxc,sigxc
c251    format (4e12.3)
        WRITE (16,250) TNAME,ITEST,stc(i),ctyc(i),IPSC(I),DEPC(I),
      1DEPCE(I)
14      CONTINUE
        xpc = 100. * xtot / q
        write (16,261) q,xtot,xpc
261     format (/'Activity released (Curies) :',f10.0,/,
      1'Total activity deposited in the TDB area (Curies) :',f12.3,/,
      2'% of activity released deposited in the TDB area  :',f12.3)
110     FORMAT (I5,38X,I5,19X,E10.4,2E11.4)
204     FORMAT (I6,30X,I4,1X,A16,15X,F7.4)
217     format (/,2x,'Table ',A2,'/',I2,'/TDB  (continued)',/)
220     FORMAT (1X,A,I7,I5,4X,A,3X,A,I2,4x,F10.1,F7.1,F9.4)
230     FORMAT (1X,A,2I7,F10.1,f4.1,f6.1)
250     FORMAT (3X,A,I7,3x,a,3x,a,I8,f12.1,f7.1)
        STOP
        END
```

ATTACHMENT A9.3 : PROGRAM CDB290.FOR

```
        PROGRAM cdb290
C
C       CALCULATES THE I-131 DEPOSITION DENSITIES NEAR NTS FROM THE
C       EXPOSURE RATES AND TOA GIVEN IN THE County DATA BASE
C       ONE PROGRAM FOR EACH TEST : CHANGE THE NAME OF THE TEST
c       prepared in mar 1990 on the basis of tdb1089.for
C       OUTPUT FILE CDB(TEST).FIN : results for the 144 sub-counties
C       OUTPUT FILE CDB(TEST).CTY : results for the 129 counties (for map
c       purposes)
C
        DIMENSION IPSC(130),NUMC(130),AREAC(130),DEPC(130)
        DIMENSION RNAME(4),RI(4),AREA(150),DEPSC(150)
```

```
      DIMENSION H12E(150),DEPCE(130),STC(130),CTYC(130)
      DIMENSION DEPSCE(150),xsc(150),npb(4)
      dimension s2xsc(150),xc(150),s2xc(150)
      REAL LON,LAT,mu,muxsc,muxc
      CHARACTER*10 TNAME,TN,RNAME
      CHARACTER*16 TT
      CHARACTER*2 ST,STC,SERIES
      CHARACTER*10 CTY,CTYC
      OPEN (UNIT=1,FILE='CDBtest.DAT', STATUS='OLD')
      OPEN (UNIT=2,FILE='cdbnum290.cty', STATUS='OLD')
      OPEN (UNIT=3,FILE='EXtest.DAT',STATUS='OLD')
      OPEN (UNIT=4,FILE='cdbloc290.crd', STATUS='OLD')
      OPEN (UNIT=15,FILE='CDBtest.FIN',STATUS='NEW')
      OPEN (UNIT=25,FILE='CDBtest.rep',STATUS='NEW')
      OPEN (UNIT=16,FILE='CDBtest.CTY',STATUS='NEW')
      OPEN (UNIT=17,FILE='check.dat',STATUS='NEW')
100   FORMAT (I4,2X,A2,1X,A10,2X,I6,I5,F10.0)
      DO 2 J = 1,129
      READ (2,100) I,STC(I),CTYC(I),IPSC(I),NUMC(I),AREAC(I)
2     CONTINUE
      DO 10 I= 1,144
      xsc(i) = 0.
      s2xsc(i) = 0.
      DEPSC(I) = 0.
      H12E(I) = 1.
10    CONTINUE
c
c     read depositions (mmCi/m2 at H+12) corresponding to a total
c     exposure rate of 1 mR/h at H+12 for Sb-131, Te-131m, Te-131,
c     and I-131 and for this particular TEST
c
      READ (3,130) TNAME,ny,nm,nd,q,SERIES,NCODE
      ITEST = ny * 10000 + nm * 100 + nd
130   FORMAT (A,3I2,e10.2,4X,A2,1X,I2)
      DO 6 J=1,4
      READ (3,140) RNAME(J),RI(J)
6     CONTINUE
140   FORMAT (A,E10.3)
c
c     calculation of the "total" I-131 deposition density per
c     unit exposure rate at H + 12
c
      ri1 = ri(1)*23.3/(60.*8.04*24.)
      ri2 = ri(2)*30.0/(8.04*24.)
      ri3 = ri(3)*25.0/(60.*8.04*24.)
      RTI= RI1 + RI2 + RI3 + RI(4)
c
c     read County Data Base data and calculate the median I-131 deposition
c     densities for all CDB areas
c
      DO 11 I = 1,144
      READ(1,110,END=3) ID,H12,H12E(ID),TOA
      RINT = RTI * EXP(-0.69315*(TOA-12.)/(8.04*24.))
      DEPSC(ID) = 1000. * H12 * RINT
      if (h12e(id).lt.1.4) h12e(id) = 1.4
      DEPSCE(ID) = H12E(ID)
      mu = log (depsc(id))
```

```
        sig = log (h12e(id))
        xsc(id) = exp (mu + (sig*sig/2.))
        s2xsc(id) = (xsc(id)**2) * (exp(sig*sig) - 1.)
11      CONTINUE
3       CONTINUE
c
c       prepare the first output file (I-131 depositions for all
c       sub-counties)
c
        write (15,219) series,ncode,tname,nm,nd,ny
        write (25,219) series,ncode,tname,nm,nd,ny
219    format(1x,'Table ',A2,'/',I2,'/CDB. ',
      4' Estimates of median I-131 depositions',
      1' per unit area of ground',/,12x,
      2' in the County Data Base area following the shot ',a,/,13x,
      3'detonated',1x,i2,'/',i2,'/19',i2,'.',/)
        write (15,228)
        write (25,228)
228     format (3x,'TEST',7x,'TEST',4x,'State',2x,'County',7x,'FIPS',
      17x,'I-131',4x,'GSD'/3x,'name',7x,'date',24x,'code',
      24x,'deposition'/
      312x,'(y/mo/d)',31x,'(nCi/m2)'/)
        l = 1
        npb(1) = 16031
        npb(2) = 35035
        npb(3) = 49252
        DO 12 I = 1,144
        READ (4,204) ID,ST,CTY,IPS1,AREA(ID)
c
c       conversion of area from ha to km2
c
        area(id) = area(id) / 100.
c
c       print the first output file (depositions for all sub-counties)
c
        WRITE (15,250) TNAME,ITEST,st,cty,IPS1,DEPSC(ID),DEPSCE(ID)
        WRITE (25,250) TNAME,ITEST,st,cty,IPS1,DEPSC(ID),DEPSCE(ID)
        if (ips1.ne.npb(l)) go to 12
        l = l + 1
        write (25,217) SERIES,NCODE
        write (25,228)
12      CONTINUE
c
c       prepare the second output file (depositions for all counties)
c
        write (16,228)
        N = 0
        xtot = 0.
        DO 14 I = 1,129
        J = NUMC(I)
        DEPC(I) = 0.
        DEPCE(I) = 1.
        xc(i) = 0.
        s2xc(i) = 0.
        muxc = 0.
        sigxc = 0.
        DO 15 K = 1,J
```

```
        N = N + 1
        XC(I) = XC(I) + (XSC(N)*AREA(N))
        xtot = xtot + (0.001 * XSC(N)*AREA(N))
        s2xc(i) = s2xc(i) + (s2xsc(n) * area(n) * area(n))
15      CONTINUE
        XC(I) = XC(I)/AREAC(I)
        s2xc(i) = s2xc(i) / (areac(i)**2)
        if (xc(i).eq.0.) go to 18
        muxc = log(xc(i)/sqrt(1. + (s2xc(i)/(xc(i)**2))))
        DEPC(I) = exp(muxc)
        sigxc = sqrt(log(1. + (s2xc(i)/(xc(i)**2))))
        DEPCE(I) = exp(sigxc)
18      continue
c       write (16,251) xtot,xc(i),s2xc(i),muxc,sigxc
c251    format (5e12.3)
        WRITE(16,250)TNAME,ITEST,stc(i),ctyc(i),IPSC(I),DEPC(I),DEPCE(I)
14      CONTINUE
        xpc = 100. * xtot / q
        write (16,261) q,xtot,xpc
261     format (/'Activity released (Curies) :',f10.0,/,
     1'Total activity deposited in the CDB area (Curies) :',f12.3,/,
     2'% of activity released deposited in the CDB area  :',f12.3)
110     FORMAT (20X,I3,17X,E8.2,2X,E9.3,3X,E8.2)
204     FORMAT (I4,2X,A2,1X,a10,2x,i6,f11.0,f7.3,f8.3)
217     format (/,2x,'Table ',A2,'/',I2,'/CDB (continued)',/)
220     FORMAT (1X,A,I7,I5,2X,A,I6,F10.1,F5.1,F8.4)
230     FORMAT (1X,A,2I7,F10.1,f4.1,f6.1)
250     FORMAT (3X,A,I7,3X,A,3X,A,I8,f12.1,f7.1)
        STOP
        END
```

ATTACHMENT A9.4. PROGRAM DEPSIMON.FOR

```
          PROGRAM DEPsimon
c
c         prepared July 1989
c         generates a single deposition file (including errors) for a given test
c
          dimension gsd(56100,13)
          DIMENSION FIPS(3100),DEP(56100,15),PRECIP(56100,15),KM(15)
          CHARACTER*12 NAMCTY
          CHARACTER*2 NAMST
          CHARACTER*8 SHOT
          INTEGER FIPS,DEP,PRECIP,GFDATE
c
c         change the names of the files + LATER CHANGES
c
       OPEN(UNIT=1,FILE = 'disk$nci:[mona.fallout]newsimonB425.FIN',
     9STATUS = 'OLD')
       OPEN(UNIT=2,FILE = 'disk$nci:[mona.fallout]newsimonK426.FIN',
     1STATUS = 'OLD')
          OPEN(UNIT=3,FILE = 'disk$nci:[mona.fallout]newsimonK427.FIN',
     2STATUS = 'OLD')
          OPEN(UNIT=4,FILE = 'disk$nci:[mona.fallout]newsimonK428.FIN',
     3STATUS = 'OLD')
          OPEN(UNIT=5,FILE = 'disk$nci:[mona.fallout]newsimonK429.FIN',
     4STATUS = 'OLD')
```

```
          OPEN(UNIT=6,FILE = 'disk$nci:[mona.fallout]newsimonK430.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=7,FILE = 'disk$nci:[mona.fallout]newsimonK501.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=8,FILE='disk$nci:[mona.fallout]newsimonH502.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=9,FILE='disk$nci:[mona.fallout]newsimonH503.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=10,FILE='disk$nci:[mona.fallout]newsimonH504.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=11,FILE='disk$nci:[mona.fallout]newsimonH505.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=12,FILE='disk$nci:[mona.fallout]newsimonH506.FIN',
     4STATUS = 'OLD')
          OPEN(UNIT=13,FILE='disk$nci:[mona.fallout]newsimonH507.FIN',
     4STATUS = 'OLD')
          OPEN (UNIT = 20, FILE = 'newsimonDEP.DAT', STATUS = 'NEW')
          OPEN (UNIT = 22, FILE = 'newsimonDEP1.DAT', STATUS = 'NEW')
          OPEN (UNIT = 21, FILE = 'newB02.DAT', STATUS = 'OLD')
          DO 1 I = 1,3094
1         READ (21,100) L,NAMST,NAMCTY,FIPS(I),DIST
100       FORMAT (I6,1X,A2,1X,A12,I6,F10.0)
C
C         JD  = NUMBER OF DAYS
C         ID = DAY (FIRST RESULT)
C         IM = MONTH (FIRST RESULT)
C         IY = YEAR
C
C
          ID = 25
          IM = 4
          IY = 53
          JD = 13
          idate = (id*10000) + (im*100) + iy
C
C
C         NO MORE CHANGES
C
C
          GFDATE = (ID*10000) + (IM*100) + IY
          DO 3 J = 1,15
          DO 3 IPS = 1,56100
          DEP(IPS,J) = 0
          PRECIP(IPS,J) = 0
3         CONTINUE
          DO 98 J = 1,JD
          READ (J,102)
          KMAX = KM(J)
          DO 4 K = 1,3094
c         DO 4 K = 1,KMAX
c         IF (J.EQ.1)READ (J,101) IPS,SHOT,IDAY,IDEP,ERR,IPRECIP
c         IF (J.GT.1)READ (J,111) IPS,SHOT,IDAY,IDEP,ERR,IPRECIP
          READ (J,111,end=99) IPS,SHOT,IDAY,IDEP,ERR,IPRECIP
          DEP(IPS,J) = IDEP
          gsd(ips,j) = exp(err)
          PRECIP(IPS,J) = IPRECIP
4         CONTINUE
```

```
99        continue
98        continue
          DO 5 I = 1,3094
          IPS = FIPS(I)
         WRITE(20,103)SHOT,idate,i,IPS,(DEP(IPS,J),gsd(ips,j),
     1PRECIP(IPS,J),J=1,6)
         WRITE(22,103)SHOT,idate,i,IPS,(DEP(IPS,J),gsd(ips,j),
     1PRECIP(IPS,J),J=7,jd)
5         CONTINUE
c101      FORMAT(I5,1X,A8,3X,I7,I8,F6.2,I4)
111       FORMAT(1X,I5,1X,A8,3X,I7,I8,F6.2,I5)
102       FORMAT (///)
c
c         jd = ??
c
103       FORMAT (1X,A,i7,i5,I6,8(I5,f5.1,I2))
          STOP
          END
```

ATTACHMENT A9.5 : CONCUST9.FOR

```
          program concustest9
c
c         prepared in nov 1991 using CONCUStest8.FOR and pasturecalc.for
c         as a basis
c         CALCULATES MILK CONCENTRATIONS AND ACTIVITY IN EACH COUNTY
c         as well as concentrations in other foodstuffs and in air
C         + average values of lumped parameters (mass interception
c         coefficient, pasture intake,..) to be used in the uncertainty
c         analysis
c
          dimension nrp(3100),picdh(100,365),picby(100,365),uncdh(100,365)
          dimension max(430),mm(430,40),reg(430),ips(56100)
          dimension ip(3100,22),ngf(3100,22),surmr(430),crsg(2)
          dimension pc(3100),vfuc(3100),ec(3100),crhc(2),crsc(2),hs(10)
          dimension surpc(3100),surpr(430),cnf(3100,6),crhg(2)
         dimension surm(3100),pi(430,50),tdiet(430),goat(3100)
          dimension ecr(430),ugf(3100,22),g(500,50),vl(500,50)
          dimension ps(9),cg1(3100,22),ntgf(3100),tcg1(3100)
          dimension cc(3100),pastday(60),nrg(3100),tmfu(3100)
          dimension cc1(3100),cc12(3100),cc2(3100),SMR(430)
          dimension tn(430),tp(430),ccr(430),cn(430),ugoat(3100)
          DIMENSION V1(430),C1(430),SRM(3100),C2(3100),cmax(3100)
          DIMENSION AC(6,3100),AR(6,430),CR(6,430),VC(6,3100),C(6,3100)
          DIMENSION VR(6,430),CT(6),VT(6),AT(6),smc(3100)
          dimension vtr(430,430),vin(430),vout(430),ain(430)
          dimension fips(3100),dist(3100),cci(3100,22),pigt(2)
          DIMENSION GFMONTH(12),NAMCTY(3100),NAMST(3100)
          dimension psw(3100),pav(3100),fav(3100),piav(3100),mfav(3100)
          dimension cmot(3100),cbc(3100),uvg(3100,22),pi2av(3100)
          dimension utgf(3100),uf(3100),umf(3100),upi(3100,22),pi2(500,50)
          dimension upi2(3100),vg(3100),bdate(430),sdate(430),uncbeg(400)
          dimension pic(3100,22),piby(3100,22),fst(3100,22),cair(3100,22)
          dimension cow(3100),cowby(3100),pcg1(3100),ucnf(3100,6)
          dimension ucow(3100),ucowby(3100),utcg(3100),ucmot(3100)
          dimension ucc(3100),ucm(3100),ucbc(3100),uc(6,3100),ucmax(3100)
c
```

```
      real mfav
      character*76 t1,t2,t3,t4,t5
      character*12 reg,namcty
      character*19 REGP
      character*2 NAMST,series
      character*15 MONTH
      CHARACTER*8 SHOw
      CHARACTER*10 SHOT
      integer fips,GFDATE,bdate,sdate
C
C     CHANGE test NAME
c
C
      open (unit=1, file = 'ps.dat', status='old')
      open (unit=5, file = '$1$dua2:[soviet.milk]newregUS.dat',
     1 status='old')
      open (unit=3, file = 'testform.dat', status='old')
      open (unit=6, file = 'testdep.dat', status='old')
      open (unit=7, file = 'newmilkdistc.dat', status='old')
      open (unit=8, file = 'pastbyc.res', status='old')
      open (unit=11, file = 'new2milktreg.dat', status='old')
      open (unit=12, file = 'newmilkdistr.dat', status='old')
      open (unit=14, file = 'newbO2.dat', status='OLD')
      open (unit=15, file = '$1$dua2:[soviet.pasture]newpastURE2.
     1 res',status='OLD')
      open (unit=16, file = '$1$dua2:[soviet.pasture]uncpi.dat',
     1 status='old')
      open (unit=17, file = 'region.def', status='old')
      open (unit=18, file = 'pastcoef.dhia', status='old')
      open (unit=19, file = 'pastcoef.byc', status='old')
      open (unit=20, file = 'pastunc.dhia', status='old')
c     open (unit=2,  file = 'testparav.dat', status='new')
      open (unit=41, file = 'testapp5.dat', status='new')
      open (unit=33, file = 'testCONC.RES', status='new')
      open (unit=13, file = 'testMILK.RES', status='new')
      open (unit=4, file = 'checkdata.dat',status='new')
c
c     END OF CHANGES
c
C
C     FIXED PARAMETER VALUES
C
c     NC = 500
      NC = 3094
      NN = 429
      UNC = 2.5
      AMBDA = 0.69315/4.5
      teff = 1. / ambda
      AMB = 0.69315/8.04
      tr = 1. / amb
      uteff = 1.3
      uteff2 = uteff*uteff
      DF1 = EXP(-0.69315*1./8.04)
      DFLV = EXP(-0.69315*0.5/8.04)
      DFGT = EXP(-0.69315*0.5/8.04)
      DFCC = EXP(-0.69315*7./8.04)
      DFGG = EXP(-0.69315*1./8.04)
```

```
      fwr = 0.5
      dfw = 0.1
      AD = 1.2
      FCC = 2.3
      pf = -0.35
      pp1 = - 0.7
      pp2 = - 0.043
      pigt(2) = 1.5
      pigt(1) = 0.00001
      crsg(2) = 0.2
      crsg(1) = 0.00001
      crwc = 75.
      crwg = 3.5
      brc = 130.
      brgt = 9.
      hw = 0.5
      prh = 0.04
      y = 0.3
      us = 1500.
      rc = 0.
      frs = 0.5
      crhc(1) = 0.1
      crhc(2) = 8.
      crhg(1) = 0.00001
      crhg(2) = 1.5
      crsc(1) = 0.3
      crsc(2) = 0.5
      hs(1) = 0.001
      hs(2) = 0.001
      hs(3) = 0.005
      hs(4) = 0.005
      hs(5) = 0.005
      hs(6) = 0.01
      hs(7) = 0.01
      hs(8) = 0.01
      hs(9) = 0.01
      utfoe = 4.
      ufc = 1.4
      ufc2 = ufc*ufc
      FGG = 1.
      ufe = 1.4
      ufe2 = ufe*ufe
      FM = 0.004
      ufmc = 2.1
      ufmc2 = ufmc*ufmc
      fmgt = 0.2
      ufmg = 2.5
      ufmg2 = ufmg*ufmg
      crmt = 0.8
      fmmt = 0.1
      TMM = crmt * fmmt
      utmm = 2.0
      utmm2 = utmm*utmm
      uflv = 2.0
      uflv2 = uflv*uflv
C
c
```

```
C         CONVERSION COEFFTS FROM klb TO kg (SIF1), FROM Mlb TO kg (SIF2)
C         AND FROM days TO years (UCF)
c
          SIF1 =  1000./2.205
          SIF2 =  1.e6/2.205
          UCF  =  1./365.
C
C         PRECIPITATION AMOUNTS CORRESPONDING TO PRECIPITATION INDICES (DRY +
C         8 CLASSES OF DAILY RAINFALL)
C         CONVERSION OF CALENDAR DATES TO JULIAN DATES FOR DEPOSITION AND PASTURE
C         INTAKE
C
          READ (1,191) T1
          READ (1,194) (PS(K),K=1,9)
          READ (1,191) T1
          READ (1,251) (GFMONTH(K),K=1,12)
          READ (1,191) T1
          READ (1,252) (PASTDAY(K), K = 1,48)
C
C         DISTANCE(KM) BETWEEN NTS AND EACH COUNTY CENTROID
C         FILE newBO2.DAT PREPARED WITH DISTANCE.FOR
C
          DO 402 I = 1,NC
          READ (14,250) L,NAMST(I),NAMCTY(I),LIPS,DIST(I),pc(i)
          IPS(LIPS) = L
          FIPS(L) = LIPS
402       CONTINUE
C
C         DEFINITION AND ORGANIZATION OF REGIONS
C         NN = NUMBER OF REGIONS
C
          do 1 n = 1,NN
          MAX(N) = 0
          read (5,10) L,reg(n),Mmax
          READ (5,20) (mm(n,m),m=1,MMAX)
          MAX(N) = MMAX
1         continue
          read (17,703)
703       format (//)
          do 701 j = 1,nc
          read (17,702) i,nrg(i),nrp(i)
          if (j.le.5) write (4,702) i,nrg(i),nrp(i)
701       continue
702       format (20x,i6,12x,i4,5x,i3)
C
C         PASTURE INTAKE VALUES FOR ALL STATES AND ALL SEASONS
C
          do 710 k = 1,70
          read (18,704) nure,bdate(nure),sdate(nure)
          read (19,705) nure
          read (20,706) nure
          na = 1
          do 707 nm = 1,48
          nb = pastday(nm)
          nx = nb - na + 1
          read (18,708) ny,(picdh(nure,nw),nw=na,nb)
          if (k.eq.1) write (4,711) nure,nm,(picdh(nure,nw),nw=na,nb)
```

```
          read (19,708) ny,(picby(nure,nw),nw=na,nb)
          read (20,709) ny,(uncdh(nure,nw),nw=na,nb)
707       na = nb + 1
710       continue
704       format (/,72x,i3,/10x,i4,10x,i4)
705       format (/,76x,i3/)
706       format (/,73x,i3/)
708       format (i5,8(1PE9.2))
709       format (i5,8f5.1)
715       format (i5,7(1PE9.2))
716       format (i5,7f5.1)
711       format (2i3,8(1PE9.2))
C
          VLMT = 0.
          ECT = 0.
          SURPT = 0.
          SURMT = 0.
          V2T = 0.
          VCFT = 0.
          VRFT = 0.
          SRMT = 0.
          VTT = 0.
          SMT = 0.
          IM = 0
C
C         READ FIPS CODES,PRECIP INDICES, AND DAILY DEPOSITION
C         RESULTS FOR EACH DAY OF THE test CONSIDERED.
C         FILE(s) CREATED BY [mona.usa]newdeptest.for
c         JD IS THE total NUMBER OF DAYS WITH DEPOSITION RESULTS
c         JD1 IS THE NUMBER OF DAYS WITH results in testdep.dat
c         JD2 IS THE NUMBER OF DAYS WITH RESULTS in testdep1.dat (if necessary)
c         JD3 IS THE NUMBER OF DAYS WITH RESULTS in testdep2.dat (if necessary)
C
          read (3,881) series,ns
881       format (8x,a2,9x,i2)
          read (3,882) shot,kd,km,iy
882       format (6x,a10,6x,3i3)
          read (3,604) jd1,jd2,jd3,jd
604       format (25x,4(5x,i2))
         IF(SERIES.eq.'RA') read (6,605)
605       format (////////////)
          do 7 i = 1,NC
         IF(SERIES.NE.'RA') GO TO 698
         read(6,60)SHOt,ID,MH,IY,M,LIPS,
     1(NGF(i,j),ugf(i,j),IP(i,j),J=1,JD1)
c
c         modif for ranger shots (shot instead of show)
c
          GO TO 7
698         read(6,60)SHOw,ID,MH,IY,M,LIPS,(NGF(i,j),ugf(i,j),IP(i,j),
     1J=1,JD1)
7         CONTINUE
          if (jd.gt.8) go to 731
          go to 739
731       continue
          open (unit=61, file = 'testdep1.dat', status='old')
          jd21 = jd1 + 1
```

```
          jd22 = jd1 + jd2
          do 732 i = 1,NC
         read(61,60)SHOw,ID,MH,IY,M,LIPS,(NGF(i,j),ugf(i,j),IP(i,j),
     1J=jd21,JD22)
732       CONTINUE
          if (jd.gt.16) go to 733
          go to 739
733       continue
          open (unit=62, file = 'testdep2.dat', status='old')
          jd31 = jd22 + 1
          do 734 i = 1,NC
         read(62,60)SHOw,ID,MH,IY,M,LIPS,(NGF(i,j),ugf(i,j),IP(i,j),
     1J=jd31,JD)
734       CONTINUE
739       continue
C
C         CG1 = CONCENTRATION IN GRASS IN A PARTICULAR DAY AND COUNTY
C         CCI = TIME-INT. CONCENTRATION IN UNDILUTED MILK (DAY,COUNTY)
C         CC =  TIME-INT. CONCENTRATION IN UNDILUTED MILK FOR THE WHOLE test
C               IN A PARTICULAR COUNTY
C
          do 12 i=1,NC
          PSW(I) = 0.
          CBC(I) = 0.
          CNF(I,6) = 0.
          UCNF(I,6) = 0.
          CC(I) = 0.
          NTGF(I) = 0
          TCG1(I) = 0.
          utgf(i) = 0.
          uf(i) = 0.
          umf(i) = 0.
          upi2(i) = 1.3
          cow(i) = 0.
          cowby(i) = 0.
          pcg1(i) = 0.
          dst = dist(i)
          VG(I) = 20150. * (DST**pf)
c
c         calculation of the median total deposition
c
          xn = 0.
          s2n = 0.
          xw= 0.
          s2w = 0.
          xgw = 0.
          sg2w = 0.
          xy = 0.
          s2y = 0.
          xp = 0.
          s2p = 0.
          xi = 0.
          s2i = 0.
c
c         kpast = 1 (off pasture); kpast = 2 (on pasture)
c
          kpast = 1
```

```
      do 412 j =1,JD
      gfdate = ID + GFMONTH(MH) + J - 1
      nure = nrp(i)
      npdb = bdate(nure)
      npde = sdate(nure)
      if ((gfdate.ge.npdb).and.(gfdate.le.npde)) kpast = 2
      pic(i,j) = picdh(nure,gfdate)
      piby(i,j) = picby(nure,gfdate)
      CAIR(i,j) = 0.
      upi(i,j) = uncdh(nure,gfdate)
      if (i.eq.1) write (4,712) i,nure,gfdate,picdh(nure,gfdate)
712   format (3i5,f6.2)
      fst(i,j) = 0.
      IF (NGF(i,j).EQ.0.) GO TO 412
      dp = ngf(i,j)
      umn = log(dp)
      usn = log(ugf(i,j))
      usn2 = usn*usn
      umsn2 = umn + (usn2/2.)
      xn = exp(umsn2) + xn
      s2n = exp(2.*umn+usn2)*(exp(usn2)-1.)+s2n
      IW = ID + J - 1
      ix = ip(i,j)
      IF (IX.LT.1) IX=1
      IF (IX.GE.5) GO TO 401
      ALPHAD = 2.8
      IF (DIST(I).LE.1540.) ALPHAD = (7.0E-04)*(DIST(I)**1.13)
      fdry = (1.-exp(-alphad*0.3))/0.3
      fstar = fdry + (3.1 - fdry)*(PS(IX)/2.5)
      GO TO 482
401   fstar = 0.9 + (11./PS(IX))
482   CONTINUE
      cg1(i,j) = ngf(i,j) * fstar
      PSW(I) = PSW(I) + NGf(i,j) * PS(IX)
      wr = 0.
      if (ps(ix).gt.0.) wr=13000.*(ps(ix)**pp1)*((dist(i)/100.)**pp2)
      CAIR(i,j) = 0.44 / (VG(I) + (WR * PS(IX) / AD))
c
c     uncertainties on CAIR - reciprocal of apparent deposition velocity
c
      if (dist(i).lt.1540.) go to 511
      uvg(i,j) = 1.5
      if (ps(ix).eq.0.) uvg(i,j) = 1.3
      go to 512
511   uvg(i,j) = 3.
      if (ps(ix).eq.0.) uvg(i,j)=2.0
512   continue
c
      fst(i,j) = fstar
c
c     uncertainties of FAV - mass interception coefficient
c
      if (dist(i).lt.1540.) go to 513
      uf(i) = 1.6
      if (ps(ix).eq.0.) uf(i)=1.3
      go to 514
513   uf(i) = 1.4
```

```
            if (ps(ix).eq.0.) uf(i) = 1.2
  514       continue
  c
            usf = log (uf(i))
            usf2 = usf*usf
            uspi = log (upi(i,j))
            uspii = uspi*uspi
            uspi2 = log (upi2(i))
            uspii2 = uspi2*uspi2
  c
  c         transfer to milk fresh from cow
  c
            p11 = fst(i,j) * teff * y * amb
            p1 = crsc(kpast) * (1. - p11) / (amb * hs(ix) * us)
            p2 = tr * (1. - p11) * rc * vg(i) * frs * teff * pic(i,j)
            p3 = brc * cair(i,j) / 0.44
            p4 = 0.001 * crwc / hw
            p5 = fst(i,j) * teff * prh * crhc(kpast)
            tfp = fst(i,j) * teff * pic(i,j)
            tfoe = p1 + p2 + p3 + p4 + p5
  c
  c         transfer to milk fresh from goat
  c
            pg1 = crsg(kpast) * (1. - p11) / (amb * hs(ix) * us)
            pg2 = tr * (1. - p11) * rc * vg(i) * frs * teff * pigt(kpast)
            pg3 = brgt * cair(i,j) / 0.44
            pg4 = 0.001 * crwg / hw
            pg5 = fst(i,j) * teff * prh * crhg(kpast)
            tfgp = fst(i,j) * teff * pigt(kpast)
            tfgoe = pg1 + pg2 + pg3 + pg4 + pg5
  c
            ust = log(uteff)
            ust2 = ust * ust
            ustfp = sqrt(usf2 + ust2 + uspii)
            ustfp2 = ustfp * ustfp
            utfp = exp(ustfp)
            uxp = log(tfp)
            uxoe = log(tfoe)
            vxp = log(tfgp)
            vxoe = log(tfgoe)
            ustfoe = log(utfoe)
            ustfoe2 = ustfoe * ustfoe
            xc = exp(uxp + 0.5*ustfp2)
            xe = exp(uxoe + 0.5*ustfoe2)
            xtfc = xc + xe
            s2tfc = xc*xc* (exp(ustfp2) - 1.) + xe*xe * (exp(ustfoe2) - 1.)
            x2c = xtfc * xtfc
            xxc = 1. + (s2tfc/x2c)
            tfc = xtfc / sqrt(xxc)
            ustfc = sqrt(log(xxc))
            ustfc2 = ustfc * ustfc
            utfc = log(xxc)
            xgp = exp(vxp + 0.5*ustfp2)
            xge = exp(vxoe + 0.5*ustfoe2)
            xtfg = xgp + xge
            s2tfg = xgp*xgp*(exp(ustfp2)-1.) + xge*xge * (exp(ustfoe2)-1.)
            x2g = xtfg * xtfg
```

```
          xxg = 1. + (s2tfg/x2g)
          tfg = xtfg / sqrt(xxg)
          ustfg = sqrt(log(xxg))
          ustfg2 = ustfg * ustfg
          utfg = log(xxg)
c
c         cowm = ngf(i,j)*fst(i,j)*pic(i,j)
          cowm = ngf(i,j) * tfc
          goatm = ngf(i,j) * tfg
          coym = ngf(i,j)*fst(i,j)*piby(i,j)
          pcgm = ngf(i,j)*fst(i,j)
          carm = ngf(i,j) * cair(i,j)
c
          l=1
       if(goatm.le.0.)write(4,4444)l,i,j,ngf(i,j),fst(i,j),pigt(kpast),
     1goatm,xgw,sg2w
          l=2
       if(coym.le.0.)write(4,4444)l,i,j,ngf(i,j),fst(i,j),piby(i,j),coym,
     1xw,s2w
          l=3
       if(pcgm.le.0.)write(4,4444)l,i,j,ngf(i,j),fst(i,j),cair(i,j),pcgm,
     1xw,s2w
          l=4
       if(carm.le.0.)write(4,4444)l,i,j,ngf(i,j),fst(i,j),cair(i,j),carm,
     1xw,s2w
          l=5
       if(uvg(i,j).le.0.)write(4,4444)l,i,j,ngf(i,j),fst(i,j),cair(i,j),
     1uvg(i,j),xw,s2w
4444      format (i2,i6,i3,i10,5e11.3)
c
c         dp = ngf(i,j)
c         umn = log(dp)
c         usn = log(ugf(i,j))
c         xn = exp(umn + (usn*usn/2.)) + xn
c         s2n = exp(2.*umn+usn*usn)*(exp(usn*usn)-1.)+s2n
          umi = log(carm)
          usi = log(uvg(i,j))
          usi2 = usi*usi + usn*usn
          xi = exp(umi + (usi2/2.)) + xi
          s2i = exp(2.*umi+usi2)*(exp(usi2)-1.)+s2i
          ucwm = log(cowm)
          ugtm = log(goatm)
          ucym = log(coym)
          ucgm = log(pcgm)
c         ucws = sqrt (usn2 + usf2 + uspii)
          ucws = sqrt (usn2 + ustfc2)
          ucws2 = ucws*ucws
                 ugts = sqrt (usn2 + ustfg2)
                 ugts2 = ugts*ugts
          ucys = sqrt (usn2 + usf2 + uspii2)
          ucys2 = ucys*ucys
          ucgs = sqrt (usn2 + usf2)
          ucgs2 = ucgs*ucgs
          xw = exp(ucwm + (ucws2/2.)) + xw
                 xgw = exp(ugtm + (ugts2/2.)) + xgw
          xy = exp(ucym + (ucys2/2.)) + xy
          xp = exp(ucgm + (ucgs2/2.)) + xp
```

```
          s2w = exp(2.*ucwm+ucws2)*(exp(ucws2)-1.)+s2w
                sg2w = exp(2.*ugtm+ugts2)*(exp(ugts2)-1.)+sg2w
          s2y = exp(2.*ucym+ucys2)*(exp(ucys2)-1.)+s2y
          s2p = exp(2.*ucgm+ucgs2)*(exp(ucgs2)-1.)+s2p
412       CONTINUE
          if (xn.eq.0.) go to 492
          x2n = xn*xn
          xx = 1. + (s2n/x2n)
c
c         modif june 1990  (nint + if test below)
c
          ntgf(i) = nint(xn/sqrt(xx))
          if (ntgf(i).eq.0.) go to 492
          usxx = sqrt(log(xx))
          utgf(i) = exp(usxx)
          x2i = xi*xi
          xxi = 1. + (s2i/x2i)
          cnf(i,6) = xi/sqrt(xxi)
          usxxi = sqrt(log(xxi))
          ucnf(i,6) = exp(usxxi)
          x2w = xw * xw
          xg2w = xgw * xgw
          x2y = xy*xy
          x2p = xp*xp
          xxw = 1. + (s2w/x2w)
          ggw = 1. + (sg2w/xg2w)
          xxy = 1. + (s2y/x2y)
          xxp = 1. + (s2p/x2p)
          cow(i) = xw / sqrt(xxw)
                 goat(i) = xgw / sqrt(ggw)
          cowby(i) = xy / sqrt(xxy)
          pcg1(i) = xp / sqrt(xxp)
          usxxw = sqrt(log(xxw))
                 usggw = sqrt(log(ggw))
          usxxy = sqrt(log(xxy))
          usxxp = sqrt(log(xxp))
          ucow(i) = exp(usxxw)
                 ugoat(i) = exp(usggw)
          ucowby(i) = exp(usxxy)
          upcg = exp(usxxp)
          upcg2 = upcg*upcg
          tcg1(i) = teff * pcg1(i)
          utcg(i) = sqrt (uteff2 + upcg2)
c         tcgm = log (tcg1(i) * 1.5)
c         tgf = ntgf(i)
c         tgfm = log (tgf * 5.8)
c         tcgu = log (utcg(i))
c         tcgu2 = tcgu*tcgu
c         tgfu = log (utgf(i))
c         tgfu2 = tgfu*tgfu
c         xa = exp (tcgm + (0.5*tcgu2))
c         xb = exp (tgfm + (0.5*tgfu2))
c         xc = xa + xb
c         sa = xa * sqrt(exp(tcgu2) - 1.)
c         sb = xb * sqrt(exp(tgfu2) - 1.)
c         xc2 = xc * xc
c         sa2 = sa * sa
```

```
c          sb2 = sb * sb
c          sc2 = sa2 + sb2
c          xxc = 1. + (sc2/xc2)
c          goam = xc/sqrt(xxc)
c          usxxc = sqrt(log(xxc))
c          goau = exp(usxxc)
c          goau2 = goau*goau
492        continue
           CNF(I,2) = fmgt * goat(i) * dfgt
           ugot2 = ugoat(i) * ugoat(i)
           ucnf(i,2)= sqrt(ufmg2 + ugot2)
           if (cnf(i,2).le.0.) ucnf(i,2) = 0.
c          cc(i) = fm * teff * cow(i)
           cc(i) = fm * cow(i)
           ucow2 = ucow(i)*ucow(i)
           ucowby2 = ucowby(i)*ucowby(i)
c          ucc(i) = sqrt (uteff2 + ufmc2 + ucow2)
           ucc(i) = sqrt (ufmc2 + ucow2)
           ucm(i) = ucc(i)
           if (cow(i).le.0.) ucm(i) = 0.
           cbc(i) = fm * teff * cowby(i) * dfgt
           ucbc(i) = sqrt (uteff2 + ufmc2 + ucowby2)
           if (cowby(i).le.0.) ucbc(i) = 0.
           cc1(i) = cc(i) * df1
           uc(1,i) = ucm(i)
           cc12(i) = cc(i) * df1 * df1
           uc(3,i) = ucm(i)
           cc2(i) = cc(i) * df1 * df1 * df1
c
c          if (i.eq.110) write (4,4445) i,fm,teff,cow(i),ucow(i),cc(i),
c     1ucm(i)
c4445      format (i6,6e12.3)
c
12         continue
C
C          MILK PRODUCTION AND DISTRIBUTION
C
C
C          REGIONAL TRANSFER IN MATRIX FORM
C          read MILKTREG.DAT (output of MILKDIStest.FOR)
c
           do 211 nd = 1,nn
           do 211 mr = 1,nn
           vtr(mr,nd) = 0.
211        continue
378        continue
           read (11,3770,end=379) nd,reg(nd),mr,reg(mr),vtr(mr,nd)
           go to 378
379        continue
           do 219 n = 1,nn
           do 219 m = 1,nn
219        vtr(n,m) = SIF1*vtr(n,m)
C
C
C          II. MILK TRANSFER BETWEEN COUNTIES OF THE SAME REGION
C
           DO 1202 I = 1,NN
```

```
          read (12,3000) n,reg(n),mmax,tp(n),tn(n),surpr(n),surmr(n)
1202      continue
          do 1201 n = 1,NC
          read (7,120) i,fips(i),nrg(i),vc(1,i),vc(2,i),vc(3,i),vc(4,i),
     1srm(i),surpc(i),surm(i),ec(i),xx,yy
1201      continue
          do 3 n = 1,NN
          cn(n) = 0.
          ccr(n) = 0.
3         continue
          do 4 n = 1,NN
          mmax = max(n)
          do 14 m = 1,mmax
          nA = mm(n,m)
          NI = IPS(NA)
          IF (SURPC(NI).GT.0.) CN(N) = CN(N) + (SURPC(NI)*CC2(NI))
14        CONTINUE
          IF (TP(N).EQ.0.) GO TO 4
          CCR(N) = CN(N)/TP(N)
4         CONTINUE
16        CONTINUE
          do 5 n = 1,NN
          mmax = max(n)
          do 6 m = 1,mmax
          nA = mm(n,m)
          NI = IPS(NA)
          C2(NI) = CCR(N)
6         CONTINUE
5         CONTINUE
C
C         III. MILK TRANSFER BETWEEN REGIONS
C
          DO 31 N = 1,NN
          V1(N) = 0.
          C1(N) = ccr(n)*df1
          vout(n) = 0.
          vin(n) = 0.
          ain(n) = 0.
31        CONTINUE
c
          do 2151 n = 1,nn
          vin(418) = vin(418) + vtr(418,N)
          vout(418) = vout(418) + vtr(n,418)
          ain(418) = ain(418) + vtr(418,N)*c1(n)
          vin(423) = vin(423) + vtr(423,N)
          vout(423) = vout(423) + vtr(n,423)
          ain(423) = ain(423) + vtr(423,N)*c1(n)
2151      continue
          v1(418) = vin(418) - vout(418)
          c1(418) = ain(418)/vin(418)
          v1(423) = vin(423) - vout(423)
          c1(423) = ain(423)/vin(423)
c
          do 214 m=1,417
          if (surpr(m).gt.0.) go to 214
          do 215 n=1,nn
          vin(m) = vin(m) + vtr(M,N)
```

```
          vout(m) = vout(m) + vtr(n,m)
          ain(m) = ain(m) + vtr(M,N)*c1(n)
215       continue
          v1(m) = vin(m) - vout(m)
          if (v1(m).eq.0.) go to 214
          if (vin(m).eq.0.) go to 214
          c1(m) = ain(m)/vin(m)
214       continue
          do 4120 m=419,422
          if (surpr(m).gt.0.) go to 4120
          do 5120 n=1,nn
          vin(m) = vin(m) + vtr(M,N)
          vout(m) = vout(m) + vtr(n,m)
          ain(m) = ain(m) + vtr(M,N)*c1(n)
5120      continue
          v1(m) = vin(m) - vout(m)
          if (v1(m).eq.0.) go to 4120
          if (vin(m).eq.0.) go to 4120
          c1(m) = ain(m)/vin(m)
4120      continue
          do 4121 m=424,nn
          if (surpr(m).gt.0.) go to 4121
          do 5121 n=1,nn
          vin(m) = vin(m) + vtr(M,N)
          vout(m) = vout(m) + vtr(n,m)
          ain(m) = ain(m) + vtr(M,N)*c1(n)
5121      continue
          v1(m) = vin(m) - vout(m)
          if (v1(m).eq.0.) go to 4121
          if (vin(m).eq.0.) go to 4121
          c1(m) = ain(m)/vin(m)
4121      continue
C
C         IV. PREPARATION OF OUTPUT DATA BY COUNTY AND REGION AS WELL
C             AS FOR THE ENTIRE AREA
C
          DO 32 N = 1,NN
          MMAX = MAX(N)
          DO 33 M = 1,MMAX
          nA = mm(n,m)
          NI = IPS(NA)
          SRM(NI) = V1(N)*SURM(NI)/SURMR(N)
c         VC(1,NI) = VCFC(NI)
c         VC(2,NI) = VRFC(NI)
c         VC(3,NI) = VLM(NI)
c         VC(4,NI) = V2(NI)
          VC(5,NI) = SRM(NI)
          VC(6,NI) = 0.
          AC(6,NI) = 0.
          C(1,NI) = CC1(NI)
          C(2,NI) = CC1(NI)
          C(3,NI) = CC12(NI)
          C(4,NI) = 0.
          C(5,NI) = 0.
          if (vc(4,ni).gt.0.) C(4,NI) = C2(NI)
          if (vc(5,ni).gt.0.) C(5,NI) = C1(N)
          DO 310 IN = 1,6
```

```
            AR(IN,N) = 0.
310         VR(IN,N) = 0.
            DO 301 IN = 1,5
            AC(IN,NI) = VC(IN,NI)*C(IN,NI)*ucf
            AC(6,NI) = AC(6,NI) + AC(IN,NI)
            VC(6,NI) = VC(6,NI) + VC(IN,NI)
301         CONTINUE
            AC(6,NI) = AC(6,NI) - AC(2,NI)
            VC(6,NI) = VC(6,NI) - VC(2,NI)
            IF (VC(6,NI).NE.0.) GO TO 654
            VC(6,NI) = 1.
654         C(6,NI) = AC(6,NI)/(VC(6,NI)*UCF)
            SMC(NI) = - EC(NI) + VC(6,NI)
            cmax(ni) = c(1,ni)
            if (c(4,ni).gt.c(1,ni)) cmax(ni) = c(4,ni)
            if (c(5,ni).gt.cmax(ni)) cmax(ni) = c(5,ni)
            IF (C(1,NI).EQ.0.) GO TO 867
            mfav(ni) = c(6,ni) / c(1,ni)
            GO TO 866
867         MFAV(NI) = 0.
866         CONTINUE
c
c           uncertainties of mfav - modifying factor for milk
c
            umf(ni)=2.
            if (mfav(ni).lt.2.) umf(ni) = 1.5
            if (mfav(ni).lt.1.1) umf(ni) = 1.1
            if (mfav(ni).lt.0.9) umf(ni) = 1.5
            if (mfav(ni).lt.0.5) umf(ni) = 2.0
            uc(4,ni) = ucc(ni)
            if (c(4,ni).le.0.) uc(4,ni) = 0.
            uc(5,ni) = ucc(ni)
            if (c(5,ni).le.0.) uc(5,ni) = 0.
            ucc2 = ucc(ni)*ucc(ni)
            umff = umf(ni)*umf(ni)
            uc(6,ni) = sqrt (ucc2 + umff)
            if (c(6,ni).le.0.) uc(6,ni) = 0.
            ucmax(ni) = uc(6,ni)
            if (cmax(ni).le.0.) ucmax(ni) = 0.
33          CONTINUE
32          CONTINUE
C
C           CALCULATION OF I-131 CONCENTRATIONS FOR THE OTHER PATHWAYS
C
            WRITE (33,331) series,ns,shot,mh,kd,iy
c
c
            i1 = 1
            i2 = 40
            NP = 0
            do 614 ij = 1,100
            NP = NP + 1
            if (i1.ge.nc) go to 615
            if (i2.ge.nc) i2 = nc
            if (ij.ne.1) WRITE (33,1331) series,ns
            WRITE (33,332)
            WRITE (33,333)
```

```
          DO 603 I = i1,i2
c
          CNF(I,1) = C(6,I)
c         IF (TCG1(I).LE.0.) GO TO 601
c         RAT = CC(I)/TCG1(I)
c         GO TO 602
c601      RAT = 0.
c602      CNF(I,2) = 0.45*(TCG1(I)*1.5 + NTGF(I)*5.8)
c         IF (RAT.LT.0.01) CNF(I,2) = CNF(I,2)/20.
          CNF(I,3) = CC(I) * fcc * dfcc
          CNF(I,4) = CC(I) * fgg * dfgg
          CNF(I,5) = tcg1(i) * fwr * dflv * dfw
c         CNF(I,5) = NTGF(I) * 1.2 * vl2
          cmot(i) = c(6,i) * tmm
          if (cmot(i).le.1.e-9) cmot(i) = 0.
          ulv2 = utcg(i) * utcg(i)
          uccc = ucm(i)*ucm(i)
          uc66 = uc(6,i)*uc(6,i)
          ucnf(i,3)=sqrt(uccc+ufc2)
          if (ucm(i).le.0.) ucnf(i,3) = 0.
          ucnf(i,4)=sqrt(uccc+ufe2)
          if (ucm(i).le.0.) ucnf(i,4) = 0.
          ucnf(i,5)=sqrt(ulv2+uflv2)
          if (utcg(i).le.0.) ucnf(i,5) = 0.
          ucmot(i)=sqrt(uc66+utmm2)
          if (cmot(i).le.0.) ucmot(i) = 0.
          write (33,330) namst(i),namcty(i),CMot(I),ucmot(i),
     1(cnf(i,k),ucnf(i,k),k=2,6)
603       CONTINUE
          i1 = i2 + 1
          i2 = i2 + 42
          WRITE (33,1332) SERIES,NS,NP
614       continue
615       continue
c
          WRITE (41,772)
          ACS=0.
c         DO 770 I = 68,120
          DO 770 I = 1,NC
          ACS = ACS + AC(6,I)
          WRITE(41,771) SHOT,IY,I,NAMST(I),NAMCTY(I),FIPS(I),NRG(I),PC(I),
c      1DIST(I),NTGF(I),TCG1(I),CC(I),C(6,I),cmax(I),AC(6,I)
     1DIST(I),NTGF(I),utgf(i),TCG1(I),utcg(i),cow(i),ucow(i),cowby(i),
     2ucowby(i)
770       CONTINUE
          WRITE (41,775) ACS
c
c
          WRITE (13,831) series,ns,shot,mh,kd,iy
          i1 = 1
          i2 = 40
          NPM = 0
          do 616 ij = 1,100
          NPM = NPM + 1
          if (i1.ge.nc) go to 617
          if (i2.ge.nc) i2 = nc
          if (ij.ne.1) WRITE (13,1831) series,ns
```

```
          WRITE (13,832)
          WRITE (13,833)
          DO 760 I = i1,i2
c
          CAV = NTGF(I) * FAV(I) * TEFF * PIAV(I) * FM * DF1 * MFAV(I)
          DIF = C(6,I) - CAV
C
          write (13,830) namst(i),namcty(i),cc(i),ucm(i),c(1,i),uc(1,i),
     1c(3,i),uc(3,i),c(4,i),uc(4,i),c(5,i),uc(5,i),c(6,i),uc(6,i),
     2cmax(i),ucmax(i),cbc(i),ucbc(i)
c         WRITE (2,782) FIPS(I),DIST(I),NTGF(I),utgf(i),FAV(I),Uf(i),
c      3PAV(I),PIAV(I),Upi(i),PI2AV(I),Upi2(i),MFAV(I),Umf(i),VGAV(I),
c      4Uvg(i)
760       continue
          i1 = i2 + 1
          i2 = i2 + 42
          WRITE (13,1832) SERIES,NS,NPM
616       continue
617       continue
c
c         do 555 i = 1,100
c         write (4,883) fips(i),(vc(k,i),k=1,6)
c         write (4,883) fips(i),(c(k,i),k=1,6)
c         write (4,883) fips(i),(ac(k,i),k=1,6)
c555      continue
c883      format (i10,6f10.0)
c
10        format (i4,1x,A12,I3)
20        format (20I6)
60        format (1x,a,1x,3i2,i5,i6,8(i5,f5.1,i2))
120       format (i5,i6,i4,4f10.0,f14.0,3f12.0,f14.0,f10.3)
191       FORMAT (1x,a76)
194       FORMAT (9F7.2)
250       FORMAT (I6,1X,A2,1X,A12,I6,F10.0,f12.0)
251       FORMAT (12F5.0)
252       FORMAT (4F5.0)
253       FORMAT (/I5,i2,A,2I3,F6.2,F5.0,F6.2,5X,F6.3,5X,F6.3,f8.2,2x,2i4)
255       FORMAT (/I5,i2,A,2I3)
254       FORMAT (2X,A,4F6.1)
771       FORMAT (1X,A10,2I5,1X,A2,1X,A12,I6,I5,F11.0,F6.0,I6,f5.1,
     13(f11.3,f5.1))
772       FORMAT(/,1X,'test',6X,'YEAR',3X,'I',' ST',' COUNTY',8X,'FIPS',
     13X,'NRG',5X,'POP.',2x,' DIST',3X,' DEP.',2X,'UNC  ',2X,
     2'GRASS    UNC',4X,'COW',4X,'UNC',4X,'COWBY   UNC'/59X,'(km)'
     3,' (nCi/m2)',6x,'(nCi.d/kg)',07X,'(nCi/d)',14X,'(nCi/d)'/)
775       FORMAT (113X,E11.3)
780       format(5x,'Best estimates and uncertainties of lumped parameter'
     1,' values for each county',
     2/,16x,' of the contiguous United States and for the shot ',A,
     3'detonated ',i2,'/',i2,'/','19',i2,'.',//)
781       FORMAT(4x,'fips,dist',3X,'ntgf,unc',5X,'fav,unc',5X,'pav',6X,
     1'piav,unc',5X,'pi2av,unc',7X,'mfav,unc',5X,'vgav,unc'/)
782       format(3x,i5,f6.0,i6,f5.1,f8.2,f6.1,f8.2,f8.1,f6.2,f8.1,f4.1,
     52(F9.1,f5.1))
830       format (3x,a,1x,a,8(1pe7.1e1,0pf5.1,2x))
1832    format(/,60X,'- A.',A,'/',I2,'/M.',I2,' -',//,1H1)
1831    format(
```

```
     11x,'TABLE ',a,'/',i2,'/M (continued)',/,1x,127('-'))
831   format(1x,'TABLE ',a,'/',i2,'/M. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 (nCi d/L) a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in all categories of cows milk considered',/,18x,'for
     4each county of the contiguous United States and for the shot ',
     5A,/,18x,'detonated ',i2,'/',i2,'/','19',i2,':',/,01x,127('-'))
832       FORMAT(61x,'Originating',3x,'Originating',/,22x,'Fresh',6x,
     1'Consumed on',4x,'Retailed',5x,'from the',6x,'from another',
     24x,'Volume-',/,
     322x,'from cow',3x,'the farm',7x,'from farm',4x,'same region',3x,
     4'region',10x,'weighted',21x,'Backyard')
833       format (1x,'State County',20x,'(category 1)',2x,'(category 2)',
     12x,'(category 3)',2x,'(category 4)',4x,'average',08x,'Maximum',
     209x,'cow',/,16x,8(5x,9('-')),/,16x,8(5x,'GM    GSD'),/,
     31x,127('-'))
330       format (3x,a,1x,a,6(1pe9.2e1,0pf5.1,4x))
1332   format(/,60X,'- A.',A,'/',I2,'/C.',I2,' -',//,1H1)
1331   format(
     11x,'TABLE ',a,'/',i2,'/C (continued)',/,01x,122('-'))
331   format(1x,'TABLE ',a,'/',i2,'/C. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in air and foodstuffs other than cows milk',/,18x,'used
     4 to calculate doses in each county',
     5' of the contiguous United States and for the shot ',A,
     6/,18x,'detonated ',i2,'/',i2,'/','19',i2,':',/,01x,122('-'))
332       FORMAT (22X,'Mothers',11x,'Goats ',
     111x,'Cottage',28x,'Leafy',13x,'Ground-level',/,22x,
     2'milk',14x,'milk',13x,'cheese',12x,'Eggs',13x,'vegetables',8x,
     3'air',/,1x,'State',1x,'County',09x,'(nCi d/L)',
     49x,'(nCi d/L)',8x,'(nCi d/kg)',8x,'(nCi d/kg)',7x,'(nCi d/kg)',
     58x,'(nCi d/m3)')
333       format (14x,6(6x,12('-')),/,14x,6(6x,'   GM    GSD'),/,
     101x,122('-'))
3000      format (1x,i3,2x,a12,i3,2x,4f14.0)
3770      format (1x,i3,2x,a12,i3,2x,a12,f10.2)
          stop
          end
```

ATTACHMENT A9.6 : DISTANCE.FOR

```
          PROGRAM DISTANCE
C
C         CALCULATES THE DISTANCE BETWEEN NTS AND EACH COUNTY CENTROID
C         PREPARED 8 JANUARY 1988 and revised 14 july 1989
C
          INTEGER FIPS
          CHARACTER*2 NAMST
          CHARACTER*12 NAMCTY
          REAL LAT,LON,LATX,LONX
          integer area,arekm
          OPEN (UNIT = 1, FILE = 'newB01.TAPE', STATUS = 'OLD')
          OPEN (UNIT = 2, FILE = 'newB01.DAT', STATUS = 'NEW')
          LAT = 37.0
          LON = 116.0
          CTE = 40000./360.
```

```
          DO 1 I = 1,3100
          READ (1,10,END=30) FIPS,NAMST,NAME,NAMCTY,area,LATX,LONX
          dat = 3.1416 * lat / 180.
          cat = cos(dat)
          DIST = CTE*SQRT((LATX-LAT)**2 + (((LONX-LON)*CAT)**2))
          arekm = area / 100
          WRITE(2,20) I,NAMST,NAMCTY,FIPS,arekm,DIST
1         CONTINUE
30        CONTINUE
10        FORMAT (I5,A2,A4,A12,6x,i10,2F8.3)
20        FORMAT (I6,1X,A2,1X,A12,I6,i8,F10.0)
          STOP
          END
```

ATTACHMENT A9.7 :NUPAST.FOR

```
          PROGRAM NUPAST
c         REVISION TO CHANGE DAILY TOTAL DRY MATTER INTAKE TO 305 DAYS
C         OF MILK (ANDRE'S CALCULATION), [AND used to CORRECT NON-NCDRPL
C         STATES TO 30% INCREASE IN PASTURE INTAKE]
C         revised 31 March 1988 After G.Wards visit.
C         PERCENT OF DIET FROM PASTURE
C
c         renamed on 4 April 1988
C         REVISED ON 9 MARCH 1989 TO NEWPASTURE.FOR
C
          DIMENSION STATE(99),IPS(99),MONTH(12),PASTPC(99,90),PASTW(99,90)
          DIMENSION TW(99),PASTAV(99),PASTDAYS(99),TPAST(99),DIET(99)
          DIMENSION seasondiet(99),DHIAPAST(99),ratio(99),spastav(99)
          dimension bdate(99),sdate(99),sdhiapast(99),sdays(99),tdate(99)
          dimension ratioday(99)
          integer bdate,sdate,pastdays,sdays,tdate
          CHARACTER*19 STATE
          Character*15 MONTH
          CHARACTER*80 T1
          OPEN (UNIT=1, FILE = 'NEWPASTure.DAT', STATUS ='OLD')
          OPEN (UNIT=2, FILE = 'NEWpasture.RES', STATUS ='NEW')
C
C         PROGRAM PREPARED BY AB JUNE 1987
C         REVISED BY MD NOVEMBER 11,1987,20 NOV 1987,18 JAN 1988,26Feb88
C         CALCULATES THE INTAKES OF PASTURE BY COW (kg/d) WEEK BY WEEK
C         AND STATE BY STATE
C         I OR M = STATE, J = MONTH, K = WEEK
C
C         USING THE YEARLY AVERAGE OF THE TOTAL DIET
C         COMPARE THE DIET(M) VALUE TO THE CALCULATED
C         DHIA %NE PASTURE (DHIAPAST(M))
C
C
          DO 70 I = 1,99
          TPAST(I) = 0.
70        CONTINUE
C
C
          READ (1,200) T1
          DO 10 I = 1,70
C         N = I
```

```
          READ (1,100) IPS(I)
          M = IPS(I)
          READ (1,120) STATE(M),PASTAV(M),spastav(m),PASTDAYS(M),
     1  sdays(m),DHIAPAST(M),sdhiapast(m),bdate(m),sdate(m)
          JA = -3
          DO 10 J = 1,12
          JA = JA + 4
          JB = JA + 3
          READ (1,110) MONTH(J),(PASTPC(M,JJ),JJ=JA,JB)
10        CONTINUE
          DO 60 I =1,70
          M = IPS(I)
          tw(m) = 0.
          IF (PASTAV(M).EQ.0.) GO TO 60
          DO 20 K = 1,48
          IF (PASTPC(M,K).GT.0.) TW(M) = TW(M) + 1.
          TPAST(M) = TPAST(M) + PASTPC(M,K)
20        CONTINUE
c to calculate the average of the pasture season
          if(tw(m).lt.1.) go to 61
          tdiet(m) = tpast(m)/48.
          seasondiet(m) = (TPAST(M)/TW(M)) * pastav(m)
          tdate(m) = sdate(m) - bdate(m)
          R1 = PASTDAYS(M)
          R2 = TDATE(M)
          ratioday(m) = R1/R2
          go to 60
61        DIET(M) = 0.
60        CONTINUE
          DO 30 I = 1,70
          M = IPS(I)
          IF (PASTAV(M).EQ.0.) GO TO 30
          DO 30 K = 1,48
          PASTW(M,K) = PASTAV(M) * PASTPC(M,K)
          IF (DIET(M).GT.0) ratio(m) = dhiapast(m)/tdiet(m)
30        CONTINUE
          DO 40 I = 1,70
          M = IPS(I)
          IF (PASTAV(M).EQ.0.) GO TO 40
        WRITE(2,130)M,STATE(M),PASTAV(M),spastav(m),DIET(M),
     1 DHIAPAST(M),sdhiapast(m),ratio(m),seasondiet(m),
     2 PASTDAYS(M),sdays(m),bdate(m),sdate(m),tdate(m),ratioday(M)
          JA = -3
          DO 50 J = 1,12
          JA = JA + 4
          JB = JA + 3
          WRITE (2,140) MONTH(J),(PASTW(M,JJ),JJ=JA,JB)
50        CONTINUE
40        CONTINUE
100       FORMAT (I5)
120       FORMAT (A,2F5.1,2I4,2F6.3,2I4)
110       FORMAT (2X,A,4F5.2)
130       FORMAT (/I3,A,2F5.1,F6.3,2F6.2,F6.3,2x,f6.3,/20x,5I4,2x,f5.2)
140       FORMAT (2X,A,4F6.1)
200       FORMAT (A)
          STOP
          END
```

ATTACHMENT A9.8 : NEWPASTREG.FOR

```
      PROGRAM NEWPASTREG
C
C     PREPARED 20 APRIL 1988
C     GENERATES ONE SET OF PASTURE INTAKES PER REGION
C     revised 10 march 1989
c     revised 06 April 1989
c     revised 23 July 1989
c
       DIMENSION REGION(100),NREG(100,11,20),MONTH(12),PI(100,50)
       DIMENSION PASTAV(100),PASTDAYS(100),DIET(100),DHIAPAST(100)
       DIMENSION RATIO(100),bdate(100),sdate(100),pastseason(100)
      dimension spastav(100),sdhiapast(100),seasondiet(100)
      dimension sdays(100),tdate(100),ratioday(100),mm(100)
      integer bdate,sdate,sdays,tdate,pastdays
      character*19 region
      CHARACTER*15 MONTH,REGNAME
      CHARACTER*80 T1,T2
C      OPEN (UNIT=1,FILE='NEWPASTREG.DEF',STATUS='OLD')
       OPEN (UNIT=1,FILE='BYCOWREG.DEF',STATUS='OLD')
       OPEN (UNIT=2,FILE='NEWPASTURE.RES',STATUS='OLD')
       OPEN (UNIT=3,FILE='NEWPASTURE2.RES',STATUS='NEW')
       OPEN (UNIT=4,FILE='NEWCHECK.RES',STATUS='NEW')
      NP=70
      NR=429
      DO 10 M=1,100
      DO 10 N=1,11
      DO 10 L=1,20
10    NREG(M,N,L)=0
      NS=0
      READ (1,400)T1
      READ (1,400) T2
400   FORMAT(A)
      DO 1 I=1,NP
      READ (1,100) M,N,MM(m),REGNAME,NMAX
      NS=NS+NMAX
      READ (1,200) (NREG(M,N,L),L=1,NMAX)
1     CONTINUE
      WRITE (4,500) NS
500   FORMAT (I10)
      DO 2 I=1,NP
      READ (2,253)M,REGION(M),PASTAV(M),spastav(m),
     1 tdiet(m),dhiapast(m),sdhiapast(m),ratio(m),seasondiet(m),
     1 PASTDAYS(M),sdays(m),bdate(m),sdate(m),tdate(m),ratioday(m)
      JA=-3
      DO 405 L=1,12
      JA = JA + 4
      JB = JA + 3
      READ (2,254) MONTH(L),(PI(M,JJ),JJ=JA,JB)
405   CONTINUE
2     CONTINUE
      DO 3 I=1,NR
      DO 4 M=1,100
      DO 4 N=1,11
      DO 4 l=1,20
      IF(NREG(M,N,L).NE.I) GO TO 4
```

```
      WRITE(3,300)NREG(M,N,L),mm(m),REGION(M),L,M,PASTAV(M),PASTDAYS(M),
     2DIET(M),DHIAPAST(M),RATIO(m),seasondiet(m),bdate(m),sdate(m)
        JA=-3
        DO 406 K=1,12
        JA = JA + 4
        JB = JA + 3
        WRITE (3,254) MONTH(K),(PI(M,JJ),JJ=JA,JB)
406     CONTINUE
        GO TO 3
4       CONTINUE
3       CONTINUE
100     FORMAT (1X,I2,I3,I2,2X,A,I3)
200     FORMAT (20I4)
253     FORMAT (/I3,A,2F5.1,F6.3,2f6.2,F6.3,2X,F6.3,/20X,5i4,2x,f5.2)
254     FORMAT (2X,A,4F6.1)
300     FORMAT (/I5,i2,A,2I3,F6.2,i5,f6.2,5X,F6.3,5X,F6.3,2x,f6.2,2x,2i4)
        STOP
        END
```

ATTACHMENT A9.9 : MILLER.FOR

```
        PROGRAM MILLER
C
C
C       SUPERSEDES OLDMILLER
C  Revised 06 JANUARY 1988 by AB - TO PROCESS SEVERAL STATES AT THE SAME
C        TIME
c
c       **10 Feb 1989 - added state 02 for subcounties in NV**
c       14 = UT; 03 = AZ; 07=CA
c        **** so some unit numbers are repeated!!!***
C       Modification in June 1989: MCFC Is always smaller than EC
c
         DIMENSION MPC(1000),RATIO1(1000),RATIO2(1000),MCFC(1000)
         DIMENSION MMC(1000),MFUC(1000),EC(1000),SC(1000),CC(1000)
         DIMENSION FC(1000),PC(1000),EXC(1000),TMFU(1000)
         DIMENSION NCM(40),NC(40,40),RPC(40),RMCFC(40),RMUFC(40)
         DIMENSION RMMC(40),RMFUC(40),REC(40),REXC(40),RTMFU(40)
         DIMENSION RSC(40),IPS(1000),RMPC(40),RATIO3(1000)
         DIMENSION MUFC(1000),IREG(60),ICYX(60)
        REAL MPC, MCFC,  MMC, MFUC, MPAVG, MCFS, MRFS, MMS,
     2  MUFS,MUFC,MPTEST
        INTEGER YEAR
C
C
C       SELECT THE STATES (BY FIPS CODE) AND THE YEAR
C       FILE UNIT NUMBERS = 1 TO 20 FOR CENSUS
C                           21 TO 40 FOR REGION
C                           41 TO 60 FOR POP
C                           61 TO 80 FOR MILLER
C                                          81 (FIXED) FOR newNUMBERS.DAT
c
c        OPEN (UNIT = 1, FILE ='CENSUS01.1954',STATUS = 'OLD')
c        OPEN (UNIT = 1, FILE ='CENSUS02.1954',STATUS = 'OLD')
cc       OPEN (UNIT = 2, FILE ='CENSUS03.1954',STATUS = 'OLD')
```

```
cc      OPEN (UNIT = 2, FILE ='CENSUS04.1954',STATUS = 'OLD')
c       OPEN (UNIT = 3, FILE ='CENSUS05.1954',STATUS = 'OLD')
cc      OPEN (UNIT = 4, FILE ='CENSUS06.1954',STATUS = 'OLD')
cc      OPEN (UNIT = 4, FILE ='CENSUS07.1954',STATUS = 'OLD')
c       OPEN (UNIT = 5, FILE ='CENSUS08.1954',STATUS = 'OLD')
c       OPEN (UNIT = 6, FILE ='CENSUS09.1954',STATUS = 'OLD')
c       OPEN (UNIT = 7, FILE ='CENSUS10.1954',STATUS = 'OLD')
c       OPEN (UNIT = 8, FILE ='CENSUS11.1954',STATUS = 'OLD')
c       OPEN (UNIT = 9, FILE ='CENSUS12.1954',STATUS = 'OLD')
c       OPEN (UNIT = 10, FILE ='CENSUS13.1954',STATUS = 'OLD')
c       OPEN (UNIT = 10, FILE ='CENSUS14.1954',STATUS = 'OLD')
c       OPEN (UNIT = 11, FILE ='CENSUS16.1954',STATUS = 'OLD')
c       OPEN (UNIT = 12, FILE ='CENSUS17.1954',STATUS = 'OLD')
c       OPEN (UNIT = 13, FILE ='CENSUS18.1954',STATUS = 'OLD')
c       OPEN (UNIT = 14, FILE ='CENSUS19.1954',STATUS = 'OLD')
c       OPEN (UNIT = 15, FILE ='CENSUS20.1954',STATUS = 'OLD')
c       OPEN (UNIT = 16, FILE ='CENSUS21.1954',STATUS = 'OLD')
c       OPEN (UNIT = 17, FILE ='CENSUS22.1954',STATUS = 'OLD')
c       OPEN (UNIT = 18, FILE ='CENSUS23.1954',STATUS = 'OLD')
c       OPEN (UNIT = 1, FILE ='CENSUS24.1954',STATUS = 'OLD')
c       OPEN (UNIT = 2, FILE ='CENSUS25.1954',STATUS = 'OLD')
c       OPEN (UNIT = 3, FILE ='CENSUS26.1954',STATUS = 'OLD')
c       OPEN (UNIT = 4, FILE ='CENSUS27.1954',STATUS = 'OLD')
c       OPEN (UNIT = 5, FILE ='CENSUS28.1954',STATUS = 'OLD')
c       OPEN (UNIT = 6, FILE ='CENSUS29.1954',STATUS = 'OLD')
c       OPEN (UNIT = 7, FILE ='CENSUS30.1954',STATUS = 'OLD')
c       OPEN (UNIT = 8, FILE ='CENSUS31.1954',STATUS = 'OLD')
cC      OPEN (UNIT = 9, FILE ='CENSUS32.1954',STATUS = 'OLD')
c       OPEN (UNIT = 10, FILE ='CENSUS33.1954',STATUS = 'OLD')
c       OPEN (UNIT = 11, FILE ='CENSUS34.1954',STATUS = 'OLD')
c       OPEN (UNIT = 12, FILE ='CENSUS35.1954',STATUS = 'OLD')
c       OPEN (UNIT = 13, FILE ='CENSUS36.1954',STATUS = 'OLD')
c       OPEN (UNIT = 14, FILE ='CENSUS37.1954',STATUS = 'OLD')
c       OPEN (UNIT = 15, FILE ='CENSUS38.1954',STATUS = 'OLD')
c       OPEN (UNIT = 16, FILE ='CENSUS39.1954',STATUS = 'OLD')
c       OPEN (UNIT = 17, FILE ='CENSUS40.1954',STATUS = 'OLD')
c       OPEN (UNIT = 18, FILE ='CENSUS41.1954',STATUS = 'OLD')
c       OPEN (UNIT = 1, FILE ='CENSUS42.1954',STATUS = 'OLD')
c       OPEN (UNIT = 2, FILE ='CENSUS44.1954',STATUS = 'OLD')
c       OPEN (UNIT = 3, FILE ='CENSUS45.1954',STATUS = 'OLD')
c       OPEN (UNIT = 4, FILE ='CENSUS46.1954',STATUS = 'OLD')
c       OPEN (UNIT = 5, FILE ='CENSUS47.1954',STATUS = 'OLD')
c       OPEN (UNIT = 6, FILE ='CENSUS48.1954',STATUS = 'OLD')
cCc     OPEN (UNIT = 7, FILE ='CENSUS49.1954',STATUS = 'OLD')
c       OPEN (UNIT = 8, FILE ='CENSUS50.1954',STATUS = 'OLD')
c       OPEN (UNIT = 9, FILE ='CENSUS51.1954',STATUS = 'OLD')
        OPEN (UNIT = 10, FILE ='CENSUS53.1954',STATUS = 'OLD')
        OPEN (UNIT = 11, FILE ='CENSUS54.1954',STATUS = 'OLD')
        OPEN (UNIT = 12, FILE ='CENSUS55.1954',STATUS = 'OLD')
        OPEN (UNIT = 13, FILE ='CENSUS56.1954',STATUS = 'OLD')
c       OPEN (UNIT =21, FILE ='REGION01.DAT',STATUS = 'OLD')
cc      OPEN (UNIT =21, FILE ='REGION02.DAT',STATUS = 'OLD')
cc      OPEN (UNIT =22, FILE ='REGION03.DAT',STATUS = 'OLD')
cc      OPEN (UNIT =22, FILE ='REGION04.DAT',STATUS = 'OLD')
c       OPEN (UNIT =23, FILE ='REGION05.dat',STATUS = 'OLD')
cc      OPEN (UNIT =24, FILE ='REGION06.DAT',STATUS = 'OLD')
cc      OPEN (UNIT =24, FILE ='REGION07.DAT',STATUS = 'OLD')
```

```
c       OPEN (UNIT =25, FILE ='REGION08.DAT',STATUS = 'OLD')
c       OPEN (UNIT =26, FILE ='REGION09.DAT',STATUS = 'OLD')
c       OPEN (UNIT =27, FILE ='REGION10.DAT',STATUS = 'OLD')
c       OPEN (UNIT =28, FILE ='REGION11.DAT',STATUS = 'OLD')
c       OPEN (UNIT =29, FILE ='REGION12.DAT',STATUS = 'OLD')
c       OPEN (UNIT =30, FILE ='REGION13.DAT',STATUS = 'OLD')
c       OPEN (UNIT =30, FILE ='REGION14.DAT',STATUS = 'OLD')
c       OPEN (UNIT =31, FILE ='REGION16.DAT',STATUS = 'OLD')
c       OPEN (UNIT =32, FILE ='REGION17.DAT',STATUS = 'OLD')
c       OPEN (UNIT =33, FILE ='REGION18.DAT',STATUS = 'OLD')
c       OPEN (UNIT =34, FILE ='REGION19.DAT',STATUS = 'OLD')
c       OPEN (UNIT =35, FILE ='REGION20.DAT',STATUS = 'OLD')
c       OPEN (UNIT =36, FILE ='REGION21.DAT',STATUS = 'OLD')
c       OPEN (UNIT =37, FILE ='REGION22.dat',STATUS = 'OLD')
c       OPEN (UNIT =38, FILE ='REGION23.DAT',STATUS = 'OLD')
c       OPEN (UNIT =21, FILE ='REGION24.DAT',STATUS = 'OLD')
c       OPEN (UNIT =22, FILE ='REGION25.DAT',STATUS = 'OLD')
c       OPEN (UNIT =23, FILE ='REGION26.DAT',STATUS = 'OLD')
c       OPEN (UNIT =24, FILE ='REGION27.DAT',STATUS = 'OLD')
c       OPEN (UNIT =25, FILE ='REGION28.DAT',STATUS = 'OLD')
c       OPEN (UNIT =26, FILE ='REGION29.DAT',STATUS = 'OLD')
c       OPEN (UNIT =27, FILE ='REGION30.DAT',STATUS = 'OLD')
c       OPEN (UNIT =28, FILE ='REGION31.DAT',STATUS = 'OLD')
cC      OPEN (UNIT =29, FILE ='REGION32.DAT',STATUS = 'OLD')
c       OPEN (UNIT =30, FILE ='REGION33.DAT',STATUS = 'OLD')
c       OPEN (UNIT =31, FILE ='REGION34.DAT',STATUS = 'OLD')
c       OPEN (UNIT =32, FILE ='REGION35.DAT',STATUS = 'OLD')
c       OPEN (UNIT =33, fILE ='REGION36.DAT',STATUS = 'OLD')
c       OPEN (UNIT =34, FiLE ='REGION37.DAT',STATUS = 'OLD')
c       OPEN (UNIT =35, FIlE ='REGION38.DAT',STATUS = 'OLD')
c       OPEN (UNIT =36, FILe ='REGION39.DAT',STATUS = 'OLD')
c       OPEN (UNIT =37, FILE ='REGION40.DAT',STATUS = 'OLD')
c       OPEN (UNIT =38, FILE ='REGION41.DAT',STATUS = 'OLD')
c       OPEN (UNIT =21, FILE ='REGION42.DAT',STATUS = 'OLD')
c       OPEN (UNIT =22, FILE ='REGION44.DAT',STATUS = 'OLD')
c       OPEN (UNIT =23, FILE ='REGION45.DAT',STATUS = 'OLD')
c       OPEN (UNIT =24, FILE ='REGION46.DAT',STATUS = 'OLD')
c       OPEN (UNIT =25, FILE ='REGION47.DAT',STATUS = 'OLD')
c       OPEN (UNIT =26, FILE ='REGION48.DAT',STATUS = 'OLD')
ccC     OPEN (UNIT =27, FILE ='REGION49.DAT',STATUS = 'OLD')
c       OPEN (UNIT =28, FILE ='REGION50.DAT',STATUS = 'OLD')
c       OPEN (UNIT =29, FILE ='REGION51.DAT',STATUS = 'OLD')
        OPEN (UNIT =30, FILE ='REGION53.DAT',STATUS = 'OLD')
        OPEN (UNIT =31, FILE ='REGION54.DAT',STATUS = 'OLD')
        OPEN (UNIT =32, FILE ='REGION55.DAT',STATUS = 'OLD')
        OPEN (UNIT =33, FILE ='REGION56.DAT',STATUS = 'OLD')
c       OPEN (UNIT =41, FILE ='POPO1T.1954',STATUS = 'OLD')
cc      OPEN (UNIT =41, FILE ='POPO2T.1954',STATUS = 'OLD')
cc      OPEN (UNIT =42, FILE ='POPO3T.1954',STATUS = 'OLD')
cc      OPEN (UNIT =42, FILE ='POPO4T.1954',STATUS = 'OLD')
c       OPEN (UNIT =43, FILE ='POPO5T.1954',STATUS = 'OLD')
cC      OPEN (UNIT =44, FILE ='POPO6T.1954',STATUS = 'OLD')
cc      OPEN (UNIT =44, FILE ='POPO7T.1954',STATUS = 'OLD')
c       OPEN (UNIT =45, FILE ='POPO8T.1954',STATUS = 'OLD')
c       OPEN (UNIT =46, FILE ='POPO9T.1954',STATUS = 'OLD')
c       OPEN (UNIT =47, FILE ='POP10T.1954',STATUS = 'OLD')
c       OPEN (UNIT =48, FILE ='POP11T.1954',STATUS = 'OLD')
```

```
c        OPEN (UNIT =49, FILE ='POP12T.1954',STATUS = 'OLD')
c        OPEN (UNIT =50, FILE ='POP13T.1954',STATUS = 'OLD')
c        OPEN (UNIT =50, FILE ='POP14T.1954',STATUS = 'OLD')
c        OPEN (UNIT =51, FILE ='POP16T.1954',STATUS = 'OLD')
c        OPEN (UNIT =52, FILE ='POP17T.1954',STATUS = 'OLD')
c        OPEN (UNIT =53, FILE ='POP18T.1954',STATUS = 'OLD')
c        OPEN (UNIT =54, FILE ='POP19T.1954',STATUS = 'OLD')
c        OPEN (UNIT =55, FILE ='POP20T.1954',STATUS = 'OLD')
c        OPEN (UNIT =56, FILE ='POP21T.1954',STATUS = 'OLD')
c        OPEN (UNIT =57, FILE ='POP22T.1954',STATUS = 'OLD')
c        OPEN (UNIT =58, FILE ='POP23T.1954',STATUS = 'OLD')
c        OPEN (UNIT =41, FILE ='POP24T.1954',STATUS = 'OLD')
c        OPEN (UNIT =42, FILE ='POP25T.1954',STATUS = 'OLD')
c        OPEN (UNIT =43, FILE ='POP26T.1954',STATUS = 'OLD')
c        OPEN (UNIT =44, FILE ='POP27T.1954',STATUS = 'OLD')
c        OPEN (UNIT =45, FILE ='POP28T.1954',STATUS = 'OLD')
c        OPEN (UNIT =46, FILE ='POP29T.1954',STATUS = 'OLD')
c        OPEN (UNIT =47, FILE ='POP30T.1954',STATUS = 'OLD')
c        OPEN (UNIT =48, FILE ='POP31T.1954',STATUS = 'OLD')
cC       OPEN (UNIT =49, FILE ='POP32T.1954',STATUS = 'OLD')
c        OPEN (UNIT =50, FILE ='POP33T.1954',STATUS = 'OLD')
c        OPEN (UNIT =51, FILE ='POP34T.1954',STATUS = 'OLD')
c        OPEN (UNIT =52, FILE ='POP35T.1954',STATUS = 'OLD')
c        OPEN (UNIT =53, FILE ='POP36T.1954',STATUS = 'OLD')
c        OPEN (UNIT =54, FILE ='POP37T.1954',STATUS = 'OLD')
c        OPEN (UNIT =55, FILE ='POP38T.1954',STATUS = 'OLD')
c        OPEN (UNIT =56, FILE ='POP39T.1954',STATUS = 'OLD')
c        OPEN (UNIT =57, FILE ='POP40T.1954',STATUS = 'OLD')
c        OPEN (UNIT =58, FILE ='POP41T.1954',STATUS = 'OLD')
c        OPEN (UNIT =41, FILE ='POP42T.1954',STATUS = 'OLD')
c        OPEN (UNIT =42, FILE ='POP44T.1954',STATUS = 'OLD')
c        OPEN (UNIT =43, FILE ='POP45T.1954',STATUS = 'OLD')
c        OPEN (UNIT =44, FILE ='POP46T.1954',STATUS = 'OLD')
c        OPEN (UNIT =45, FILE ='POP47T.1954',STATUS = 'OLD')
c        OPEN (UNIT =46, FILE ='POP48T.1954',STATUS = 'OLD')
cC       OPEN (UNIT =47, FILE ='POP49T.1954',STATUS = 'OLD')
c        OPEN (UNIT =48, FILE ='POP50T.1954',STATUS = 'OLD')
c        OPEN (UNIT =49, FILE ='POP51T.1954',STATUS = 'OLD')
         OPEN (UNIT =50, FILE ='POP53T.1954',STATUS = 'OLD')
         OPEN (UNIT =51, FILE ='POP54T.1954',STATUS = 'OLD')
         OPEN (UNIT =52, FILE ='POP55T.1954',STATUS = 'OLD')
         OPEN (UNIT =53, FILE ='POP56T.1954',STATUS = 'OLD')
c        OPEN (UNIT =61, FILE ='[.dist]newmiller01.1954',STATUS = 'NEW')
cc        OPEN (UNIT =61, FILE ='[.dist]newMILLER02.1954',STATUS = 'NEW')
cc        OPEN (UNIT =62, FILE ='[.dist]newMILLER03.1954',STATUS = 'NEW')
cc        OPEN (UNIT =62, FILE ='[.dist]newmiller04.1954',STATUS = 'NEW')
c         OPEN (UNIT =63, FILE ='[.dist]newmiller05.1954',STATUS = 'NEW')
c        OPEN (UNIT =64, FILE ='[.dist]newmiller06.1954',STATUS = 'NEW')
cc        OPEN (UNIT =64, FILE ='[.dist]newMILLER07.1954',STATUS = 'NEW')
c         OPEN (UNIT =65, FILE ='[.dist]newmiller08.1954',STATUS = 'NEW')
c         OPEN (UNIT =66, FILE ='[.dist]newmiller09.1954',STATUS = 'NEW')
c        OPEN (UNIT =67, FILE ='[.dist]newmiller10.1954',STATUS = 'NEW')
c        OPEN (UNIT =68, FILE ='[.dist]newmiller11.1954',STATUS = 'NEW')
c         OPEN (UNIT =69, FILE ='[.dist]newmiller12.1954',STATUS = 'NEW')
c         OPEN (UNIT =70, FILE ='[.dist]newmiller13.1954',STATUS = 'NEW')
c          OPEN (UNIT =70, FILE ='[.dist]newMILLER14.1954',STATUS = 'NEW')
c         OPEN (UNIT =71, FILE ='[.dist]newmiller16.1954',STATUS = 'NEW')
```

```
c         OPEN (UNIT =72, FILE ='[.dist]newmiller17.1954',STATUS = 'NEW')
c         OPEN (UNIT =73, FILE ='[.dist]newmiller18.1954',STATUS = 'NEW')
c         OPEN (UNIT =74, FILE ='[.dist]newmiller19.1954',STATUS = 'NEW')
c         OPEN (UNIT =75, FILE ='[.dist]newmiller20.1954',STATUS = 'NEW')
c         OPEN (UNIT =76, FILE ='[.dist]newmiller21.1954',STATUS = 'NEW')
c         OPEN (UNIT =77, FILE ='[.dist]newmiller22.1954',STATUS = 'NEW')
c         OPEN (UNIT =78, FILE ='[.dist]newmiller23.1954',STATUS = 'NEW')
c         OPEN (UNIT =61, FILE ='[.dist]newmiller24.1954',STATUS = 'NEW')
c         OPEN (UNIT =62, FILE ='[.dist]newmiller25.1954',STATUS = 'NEW')
c         OPEN (UNIT =63, FILE ='[.dist]newmiller26.1954',STATUS = 'NEW')
c          OPEN (UNIT =64, FILE ='[.dist]newmiller27.1954',STATUS = 'NEW')
c          OPEN (UNIT =65, FILE ='[.dist]newmiller28.1954',STATUS = 'NEW')
c          OPEN (UNIT =66, FILE ='[.dist]newmiller29.1954',STATUS = 'NEW')
c         OPEN (UNIT =67, FILE ='[.dist]newmiller30.1954',STATUS = 'NEW')
c         OPEN (UNIT =68, FILE ='[.dist]newmiller31.1954',STATUS = 'NEW')
cC         OPEN (UNIT =69, FILE ='[.dist]newmiller32.1954',STATUS = 'NEW')
c         OPEN (UNIT =70, FILE ='[.dist]newmiller33.1954',STATUS = 'NEW')
c         OPEN (UNIT =71, FILE ='[.dist]newmiller34.1954',STATUS = 'NEW')
c         OPEN (UNIT =72, FILE ='[.dist]newmiller35.1954',STATUS = 'NEW')
c         OPEN (UNIT =73, FILE ='[.dist]newmiller36.1954',STATUS = 'NEW')
c         OPEN (UNIT =74, FILE ='[.dist]newmiller37.1954',STATUS = 'NEW')
c         OPEN (UNIT =75, FILE ='[.dist]newmiller38.1954',STATUS = 'NEW')
c         OPEN (UNIT =76, FILE ='[.dist]newmiller39.1954',STATUS = 'NEW')
c         OPEN (UNIT =77, FILE ='[.dist]newmiller40.1954',STATUS = 'NEW')
c         OPEN (UNIT =78, FILE ='[.dist]newmiller41.1954',STATUS = 'NEW')
c         OPEN (UNIT =61, FILE ='[.dist]newmiller42.1954',STATUS = 'NEW')
c         OPEN (UNIT =62, FILE ='[.dist]newmiller44.1954',STATUS = 'NEW')
c         OPEN (UNIT =63, FILE ='[.dist]newmiller45.1954',STATUS = 'NEW')
c         OPEN (UNIT =64, FILE ='[.dist]newmiller46.1954',STATUS = 'NEW')
c         OPEN (UNIT =65, FILE ='[.dist]newmiller47.1954',STATUS = 'NEW')
c          OPEN (UNIT =66, FILE ='[.dist]newmiller48.1954',STATUS = 'NEW')
ccC          OPEN (UNIT =67, FILE ='[.dist]newmiller49.1954',STATUS = 'NEW')
c         OPEN (UNIT =68, FILE ='[.dist]newmiller50.1954',STATUS = 'NEW')
c          OPEN (UNIT =69, FILE ='[.dist]newmiller51.1954',STATUS = 'NEW')
         OPEN (UNIT =70, FILE ='[.dist]newmiller53.1954',STATUS = 'NEW')
         OPEN (UNIT =71, FILE ='[.dist]newmiller54.1954',STATUS = 'NEW')
          OPEN (UNIT =72, FILE ='[.dist]newmiller55.1954',STATUS = 'NEW')
         OPEN (UNIT =73, FILE ='[.dist]newmiller56.1954',STATUS = 'NEW')
         OPEN (UNIT = 81, FILE ='newNUMBERS.DAT',STATUS = 'OLD')
c         OPEN (UNIT = 81, FILE ='NUMBERS.DAT',STATUS = 'OLD')
         OPEN (UNIT = 82, FILE ='CHECKS.DAT',STATUS = 'NEW')
C
C
C
C       INDICATE THE NUMBER OF STATES YOU WANT TO RUN (UP TO TWENTY)
C       ACCORDING TO THE FILE UNIT NUMBERS FOR CENSUS
C       NS1 = FIRST STATE, NS2 = LAST STATE
C
C
        NS1 =10
        NS2 =13
c
C        NO MORE CHANGES
C
C
        DO 600 J = 1,49
        READ  (81,601) JST,JREG,JCYX
```

```
        IREG(JST) = JREG
        ICYX(JST) = JCYX
600     CONTINUE
601     FORMAT (3I3)
        DO 602 IJK = NS1,NS2
        IR = IJK + 20
        IP = IJK + 40
        IM = IJK + 60
        READ (IJK,200) ITATE,YEAR,MPAVG,FS,PS,MCFS,MRFS,MMS,CONS,MUFS
        WRITE (82,200) ITATE,YEAR,MPAVG,FS,PS,MCFS,MRFS,MMS,CONS,MUFS
        NUMREG = IREG(ITATE)
        NCMAX = ICYX(ITATE)
        WRITE (82,197) ITATE,NUMREG,NCMAX
        PSS = 0.
        CSS = 0.
        fss = 0.
        CSM = 0.
        L=0
        mcfs = mcfs * 1000.
        mufs = mufs * 1000.
        mrfs = mrfs * 1000.
        mms = mms * 1000.
        READ (IP,402)
C       WRITE(82,402)
 10      L = L+1
         READ (IJK,400,END=301) IPS(L),CCL,FCL
         WRITE (82,400) IPS(L),CCL,FCL
        READ (IP,401,END=301) IYR,IST,ICT,NPCL
        write (82,401) iyr,ist,ict,npcl
        PCL = NPCL
        I = IPS(L)
        CC(I) = CCL
        FC(I) = FCL
        PC(I) = PCL
        PSS = PSS + PC(I)
        fss = fss + fc(i)
        CSS = CSS + CC(I)
         GO TO 10
301      CONTINUE
        N = L-1
        WRITE (82,999) N
        DO 40 L = 1,N
        I = IPS(L)
        RATIO1(I) = 0.
        RATIO3(I) = 0.
        MPC(I) = (CC(I) * MPAVG)/1000
        EC(I) =  (CONS * PC(I))/1000
        IF (FSS.GT.0.) RATIO1(I) = FC(I)/FSS
        IF (CSS.GT.0.) RATIO3(I) = CC(I)/CSS
        RATIO2(I) = PC(I)/PSS
        MCFC(I) = MCFS * RATIO1(I)
        MUFC(I) = MUFS * RATIO3(I)
        EXC(I) = MPC(I) - MUFC(I) - EC(I)
         MPTEST = (MPAVG * CSS)/1000
        IF (EXC(I).GT.0.) CSM = CSM + CC(I)
 40     CONTINUE
        DO 30 L = 1,N
```

```
        I = IPS(L)
        MMC(I) = 0.
        IF (EXC(I).GT.0.) MMC(I) = MMS * CC(I) / CSM
c
c       in case MCFC is greater than EC
c
        if (mcfc(i).gt.ec(i)) go to 3001
        MFUC(I) = MPC(I) - MUFC(I) - MCFC(I) - MMC(I)
        TMFU(I) = MPC(I) - MMC(I) - MUFC(I)
        go to 3002
3001    mfuc(i) = (mpc(i)-mufc(i)-mcfc(i)-mmc(i)) + (mcfc(i)-ec(i))
        TMFU(I) = MPC(I) - MMC(I) - MUFC(I)  + (mcfc(i)-ec(i))

        mcfc(i) = ec(i)
3002    SC(I) = TMFU(I) - EC(I)
   30   CONTINUE
        iyr = iyr + 1900
        WRITE (IM,100) ITATE, YEAR,iyr
        WRITE (IM,101)MPAVG,FS,fss,css,PS,PSS,MPTEST,MCFS,MUFS,MRFS,MMS,CONS
        WRITE (IM,501)
        WRITE (IM,102)
        TPC = 0.
        TMCFC = 0.
        TMUFC = 0.
        TMPC = 0.
        TMMC = 0.
        TMFUC = 0.
        TEC = 0.
        TEXC = 0.
        TTMFU = 0.
        TSC = 0.
        DO 15 L=1,N
        II = (Itate * 1000) + IPS(L)
        I = IPS(L)
        WRITE (IM,103) II , PC(I), MCFC(I), MUFC(I),
      2  MPC(I),MMC(I), MFUC(I), EC(I), EXC(I), TMFU(I), SC(I)
        TPC = TPC + PC(I)
        TMCFC = TMCFC + MCFC(I)
         TTMUFC = TMUFC + MUFC(I)
        TMPC = TMPC + MPC(I)
        TMMC = TMMC + MMC(I)
        TMFUC = TMFUC + MFUC(I)
        TEC = TEC + EC(I)
        TEXC = TEXC + EXC(I)
        TTMFU = TTMFU + TMFU(I)
        TSC = TSC + SC(I)
   15    CONTINUE
        WRITE (IM,104) TPC,TMCFC,TMUFC,TMPC,TMMC,TMFUC,TEC,TEXC,TTMFU,
      1TSC
        WRITE(82,195)NUMREG,NCMAX
        READ(IR,105) (NCM(J),J=1,NUMREG)
C
        DO 20 J=1,NUMREG
        RPC(J)= 0.
        RMCFC(J)=0.
        RMUFC(J)=0.
        RMPC(J)=0.
```

```
        RMMC(J)=0.
        RMFUC(J) = 0.
        REC(J) = 0.
        REXC(J) = 0.
        RTMFU(J) = 0.
        RSC(J) = 0.
        NCMAX = NCM(J)
        READ(IR,106)(NC(J,K),K=1,NCMAX)
        WRITE(82,196)NCMAX,(NC(J,K),K=1,NCMAX)
20      CONTINUE
C
         DO 21 J=1,NUMREG
        NM=NCM(J)
C
        DO 21 K=1,NM
        L=NC(J,K)
        RPC(J)=RPC(J) + PC(L)
        RMCFC(J)=RMCFC(J) + MCFC(L)
        RMUFC(J)=RMUFC(J) + MUFC(L)
        RMPC(J)=RMPC(J) + MPC(L)
        RMMC(J)=RMMC(J) + MMC(L)
        RMFUC(J) = RMFUC(J) + MFUC(L)
        REC(J) = REC(J) + EC(L)
        REXC(J) = REXC(J) + EXC(L)
        RTMFU(J) = RTMFU(J) + TMFU(L)
        RSC(J) = RSC(J) + SC(L)
   21    CONTINUE
C
        DO 22 J = 1,NUMREG
        WRITE (IM,107)
        NM = NCM(J)
C
        DO 23 K=1,NM
        L=NC(J,K)
        LL = (ITATE * 1000) + L
   23   WRITE(IM,109)LL,PC(L),MCFC(L),MUFC(L),MPC(L),MMC(L),MFUC(L),EC(L)
      3,EXC(L),TMFU(L),SC(L)
        WRITE (IM,108)
   22    WRITE (IM,109) NM,RPC(J),RMCFC(J),RMUFC(J),RMPC(J),RMMC(J),
      1RMFUC(J),REC(J),REXC(J),RTMFU(J), RSC(J)
602     CONTINUE
402     FORMAT(//)
400     FORMAT (I4,2F10.0)
401     FORMAT (3I3,I12)
  200    FORMAT (I4,I5,F6.0,F8.0,F10.0,F8.0,F7.0,F10.1,F7.2,F8.0)
  201    FORMAT (I4,4F10.0)
  300    FORMAT (3A4,I4)
  100    FORMAT (10X,'**MILLER METHOD - MILK DISTRIBUTION**',2x,
     1'check that the TEST values are the same as the estimates'//10X,
     2  'STATE',I3,30X,'YEAR',I5,2x,'TEST =',i5/)
  101    FORMAT (10X,'AVERAGE MILK PER COW (LBS/YR) = ',F10.0/,
     2  10X,'TOTAL DAIRY FARMS = ',F10.0,'       TEST = ',F10.0/,
     9  10x,'TOTAL DAIRY COWS = ',f10.0/,
     3  10X,'TOTAL POPULATION = ',F10.0,'        TEST = ',F10.0/,
     8  10X,'TOTAL MILK PROD (Klb)- check mpc total = ',F10.0/,
     4  10X,'FARM CONSUMPTION (Klb) = ',F10.0/,
     1  10X,'USED ON FARM (Klb) = ',F10.0/,
```

```
     5  10X,'FARM RETAIL (Klb) = ',F10.0/,
     6  10X,'MANUFACTURED (Klb) = ',F10.0/,
     7  10X,'CONSUMPTION RATE (lbs per capita per year) = ',F10.2)
  501   FORMAT (///30X,'PC = POPULATION '/,
     2  30X,'MCFC = MILK CONSUMED ON FARMS'/,
     3  30X,'MUFC = MILK USED ON FARMS (non-consumption by people)'/,
     4  30X,'MPC  = TOTAL MILK PRODUCED (calculated)'/,
     5  30X,'MMC  = MILK USED FOR MANUFACTURING'/,
     6  30X,'MFUC = MILK TO SELL(does not include farm consumption)-urban
     1 fluid use'/,
     7  30x,'EC   = EXPECTED CONSUMPTION (calculated)'/,
     8  30X,'EXC  = TEST TO SITE MANUFACTURING PLANTS'/,
     9  30X,'TMFU = TOTAL FLUID MILK CONSUMED'/,
    10 30X,'SURP = SURPLUS'//)
  102   FORMAT (//3X,'CNTY',6X,'PC',8X,'MCFC',8X,'MUFC',8X,'MPC',8X,
     2  'MMC',10X,'MFUC',10X,'EC',9X,'EXC',8X,'TMFU',8X,'SURP'/)
  103   FORMAT (I6,10F12.0)
  104  FORMAT (/,6X,10F12.0)
  105   FORMAT(25I3)
  106   FORMAT (15I4)
  195   FORMAT(2I20)
  196   FORMAT (I8,15I4)
  197   FORMAT (6I10)
  107   FORMAT (/,3X,'CNTY',6X,'PC',8X,'MCFC',8X,'MUFC',8X,'MPC',8X,
     2  'MMC',10X,'MFUC',10X,'EC',9X,'EXC',8X,'TMFU',8X,'SURP'/)
  108   FORMAT(/)
  109   FORMAT (I6,10F12.0)
999    FORMAT (' N =',I3)
        STOP
        END
```

ATTACHMENT A9.10 : NEWMILLERUS2.FOR

```
       PROGRAM NEWMILLERUS2
C
C      ADD THE RESULTS OBTAINED FOR THE 49 STATES
C      PREPARED 13 JANUARY 1988 (MILLERUS)
C      REVISED 18 NOVEMBER 1988 (MILLERUS2)
C      REVISED  2 MARCH 1989 (NEWMILLERUS2)
C
        DIMENSION MPC(3100),MCFC(3100),KIPS(3100),TOT(20),TOTS(10)
        DIMENSION MMC(3100),MFUC(3100),EC(3100),SC(3100),CC(3100)
        DIMENSION FC(600),PC(3100),EXC(3100),TMFU(3100),MUFC(3100)
        DIMENSION NCM(430),RPC(430),RMCFC(430),RMUFC(430),totc(10)
        DIMENSION RMMC(430),RMFUC(430),REC(430),REXC(430),RTMFU(430)
        DIMENSION RSC(430),RMPC(430),NCTY(600),region(60)
        DIMENSION REG(430),MAX(600),IST(430),NREG(430,60)
        DIMENSION NST(60,40),IPS(430,60),MM(430,60),NRT(430)
c
       dimension it(60),totr(60,6),tott(6)
c
       CHARACTER*12 REG,REGIN,region
       REAL MPC, MCFC,  MMC, MFUC, MPAVG, MCFS, MRFS, MMS,MSC,
     2  MUFS,MUFC,MPTEST,MCFCL,MUFCL,MPCL,MMCL,MFUCL
        INTEGER YEAR
c
```

```
c       MILKCTY3, MILKREG3, MILKST3 FOR THE TOTAL OVER ALL STATES
c
         OPEN (UNIT=91,FILE='NEWREGIONUS.DAT',STATUS = 'OLD')
         OPEN (UNIT=92,FILE='NEWMILKCTY3.DAT',STATUS = 'NEW')
         OPEN (UNIT=93,FILE='NEWMILKST3.DAT',STATUS = 'NEW')
         OPEN (UNIT=94,FILE='NEWMILKREG3.DAT',STATUS = 'NEW')
C        OPEN (UNIT = 81, FILE ='NEWNUMBERS.DAT',STATUS = 'OLD')
         OPEN (UNIT = 82, FILE ='CHECKS.DAT',STATUS = 'NEW')
C
C       NST(N,J) = NS = REGION CODE (REGION J IN STATE N)
C       NRT(N) = NR = NUMBER OF REGIONS IN STATE N
C       NREG(N,J) = NM = NUMBER OF COUNTIES IN REGION J OF STATE N
C       NCTOT = TOTAL NUMBER OF COUNTIES = 3094
C       NRTOT = TOTAL NUMBER OF REGIONS = 429
C
        DO 6021 K = 1,10
        TOT(K) = 0.
        TOTS(K) = 0.
        TOTC(K) = 0.
6021    CONTINUE
C
c
        do 6121 im=1,60
        do 6121 k=1,6
        totr(im,k)=0.
        tott(k)=0.
6121    continue
c
        write (94,5010)
        write (94,501)
        write (94,5011)
        write (94,1020)
        write (93,931)
        write (93,932)
        write (93,933)
        write (93,934)
c
        DO 1 N = 1,49
        NCTY(N) = 0
        READ (91,199) KST,NR,NMAX,REGIN
        READ (91,110)(NST(N,J),J=1,NR)
        READ (91,140)(NREG(N,J),J=1,NR)
        NRT(N) = NR
        region(kst) = regin
        DO 3 I = 1,NR
        NS = NST(N,I)
        REG(NS) = REGIN
        IST(NS) = KST
        JJ = NREG(N,I)
        NCTY(N) = NCTY(N) + JJ
        MAX(NS) = JJ
        READ (91,120) (IPS(NS,J),J = 1,JJ)
        DO 3 J = 1,JJ
        MM(NS,J) = (IST(NS)*1000) + IPS(NS,J)
3       CONTINUE
1       CONTINUE
199     FORMAT (3I3,1X,A12)
```

```
120     FORMAT (15I4)
110     FORMAT (18I4)
140     FORMAT (18I3)
C
C       RUN FILES IN TWO PARTS (TOP AND BOTTOM HALF OF THE STATES)
C

        OPEN (UNIT =1, FILE ='[.dist]newMILLER01.1954',STATUS = 'OLD')
C       OPEN (UNIT =2, FILE ='[.dist]newMILLER04.1954',STATUS = 'OLD')
        OPEN (UNIT =2, FILE ='[.dist]newMILLER03.1954',STATUS = 'OLD')
        OPEN (UNIT =3, FILE ='[.dist]newMILLER05.1954',STATUS = 'OLD')
C       OPEN (UNIT =4, FILE ='[.dist]newMILLER06.1954',STATUS = 'OLD')
        OPEN (UNIT =4, FILE ='[.dist]newMILLER07.1954',STATUS = 'OLD')
cc      OPEN (UNIT =5, FILE ='[.dist]newMILLER08.1954',STATUS = 'OLD')
        OPEN (UNIT =5, FILE ='[.dist]MILLER78908.1954',STATUS = 'OLD')
         OPEN (UNIT =6, FILE ='[.dist]newMILLER09.1954',STATUS = 'OLD')
        OPEN (UNIT =7, FILE ='[.dist]newMILLER10.1954',STATUS = 'OLD')
        OPEN (UNIT =8, FILE ='[.dist]newMILLER11.1954',STATUS = 'OLD')
        OPEN (UNIT =9, FILE ='[.dist]newMILLER12.1954',STATUS = 'OLD')
         OPEN (UNIT =10, FILE ='[.dist]newMILLER13.1954',STATUS = 'OLD')
cc       OPEN (UNIT =11, FILE ='[.dist]newMILLER16.1954',STATUS = 'OLD')
         OPEN (UNIT =11, FILE ='[.dist]MILLER78916.1954',STATUS = 'OLD')
         OPEN (UNIT =12, FILE ='[.dist]newMILLER17.1954',STATUS = 'OLD')
         OPEN (UNIT =13, FILE ='[.dist]newMILLER18.1954',STATUS = 'OLD')
         OPEN (UNIT =14, FILE ='[.dist]newMILLER19.1954',STATUS = 'OLD')
         OPEN (UNIT =15, FILE ='[.dist]newMILLER20.1954',STATUS = 'OLD')
cc       OPEN (UNIT =16, FILE ='[.dist]newMILLER21.1954',STATUS = 'OLD')
         OPEN (UNIT =16, FILE ='[.dist]MILLER78921.1954',STATUS = 'OLD')
         OPEN (UNIT =17, FILE ='[.dist]newMILLER22.1954',STATUS = 'OLD')
         OPEN (UNIT =18, FILE ='[.dist]newMILLER23.1954',STATUS = 'OLD')
         OPEN (UNIT =19, FILE ='[.dist]newMILLER24.1954',STATUS = 'OLD')
cc       OPEN (UNIT =20, FILE ='[.dist]newMILLER25.1954',STATUS = 'OLD')
         OPEN (UNIT =20, FILE ='[.dist]MILLER78925.1954',STATUS = 'OLD')
         OPEN (UNIT =21, FILE ='[.dist]newMILLER26.1954',STATUS = 'OLD')
         OPEN (UNIT =22, FILE ='[.dist]newMILLER27.1954',STATUS = 'OLD')
         OPEN (UNIT =23, FILE ='[.dist]newMILLER28.1954',STATUS = 'OLD')
cc       OPEN (UNIT =24, FILE ='[.dist]newMILLER29.1954',STATUS = 'OLD')
         OPEN (UNIT =24, FILE ='[.dist]MILLER78929.1954',STATUS = 'OLD')
         OPEN (UNIT =25, FILE ='[.dist]newMILLER30.1954',STATUS = 'OLD')
C
C       ENTER UNIT FILE NAMES FOR FIRST AND LAST STATE TO CONSIDER
C       1 TO 25 FOR TOP HALF
C
        DO 502 IM = 1,25
        READ (IM,100) ITATE, YEAR
        READ (IM,101)MPAVG,FS,fss,css,PS,PSS,MPTEST,MCFS,MUFS,MRFS,MMS,CONS
        READ (IM,501)
        READ (IM,102)
        IJ = 0
        it(im) = itate
        NII = NCTY(IM)
        DO 503 II = 1,NII
        I = II + IJ
        READ (IM,103,ERR=504) KIPS(I) , PC(I), MCFC(I), MUFC(I),
     2  MPC(I),MMC(I), MFUC(I), EC(I), EXC(I), TMFU(I), SC(I)
        WRITE (92,103) KIPS(I) , PC(I), MCFC(I), MUFC(I),
     2  MPC(I),MMC(I), MFUC(I), EC(I), EXC(I), TMFU(I), SC(I)
```

```
        totc(1) = totc(1) + pc(i)
        totc(2) = totc(2) + mcfc(i)
        totc(3) = totc(3) + mufc(i)
        totc(4) = totc(4) + mpc(i)
        totc(5) = totc(5) + mmc(i)
        totc(6) = totc(6) + mfuc(i)
        totc(7) = totc(7) + ec(i)
        totc(8) = totc(8) + exc(i)
        totc(9) = totc(9) + tmfu(i)
        totc(10) = totc(10) + sc(i)
c
        totr(im,3) = totr(im,3) + sc(i)
c
503     CONTINUE
504     IJ = I-1
        READ (IM,104) TPC,TMCFC,TMUFC,TMPC,TMMC,TMFUC,TEC,TEXC,TTMFU,
      1TSC
        TOTS(1) = TOTS(1) + TPC
        TOTS(2) = TOTS(2) + TMCFC
        TOTS(3) = TOTS(3) + TMUFC
        TOTS(4) = TOTS(4) + TMPC
        TOTS(5) = TOTS(5) + TMMC
        TOTS(6) = TOTS(6) + TMFUC
        TOTS(7) = TOTS(7) + TEC
        TOTS(8) = TOTS(8) + TEXC
        TOTS(9) = TOTS(9) + TTMFU
        TOTS(10) = TOTS(10) + TSC
c
        totr(im,1) = tsc
c
        ACONS = TEC*(1.E6)/(2.205*365.*TPC)
        WRITE (93,114) region(itate),ITATE,TPC,TMCFC,TMUFC,TMPC,TMMC,
      1TMFUC,TEC,TEXC,TTMFU,TSC,ACONS
C
        NR = NRT(IM)
        DO 522 J = 1,NR
        READ (IM,107)
        NM = NREG(IM,J)
C
        DO 523 K=1,NM
         READ(IM,109)LL,PCL,MCFCL,MUFCL,MPCL,MMCL,MFUCL,ECL
      3,EXCL,TMFUL,SCL
C        WRITE(94,109)NM,PCL,MCFCL,MUFCL,MPCL,MMCL,MFUCL,ECL
C     3,EXCL,TMFUL,SCL
523     CONTINUE
        READ (IM,108)
         READ (IM,109)NPQ,RPCJ,RMCFCJ,RMUFCJ,RMPCJ,RMMCJ,
      1RMFUCJ,RECJ,REXCJ,RTMFUJ, RSCJ
        L = NST(im,J)
C       WRITE  (94,998) L
        RPC(L) = RPCJ
        RMCFC(L) = RMCFCJ
        RMUFC(L) = RMUFCJ
        RMPC(L) = RMPCJ
        RMMC(L) = RMMCJ
        RMFUC(L) = RMFUCJ
        REC(L) = RECJ
```

```
        REXC(L) = REXCJ
        RTMFU(L) = RTMFUJ
        RSC(L) = RSCJ

c
        totr(im,2) = totr(im,2) + rsc(l)
c
         WRITE (94,1090) reg(l),itate,L,RPC(L),RMCFC(L),RMUFC(L),RMPC(L),
      1RMMC(L),RMFUC(L),REC(L),REXC(L),RTMFU(L), RSC(L)
522     CONTINUE
        close (im)
502     CONTINUE
C
cc       OPEN (UNIT =26, FILE ='[.dist]newMILLER31.1954',STATUS = 'OLD')
         OPEN (UNIT =26, FILE ='[.dist]MILLER78931.1954',STATUS = 'OLD')
C        OPEN (UNIT =27, FILE ='[.dist]newMILLER32.1954',STATUS = 'OLD')
         OPEN (UNIT =27, FILE ='[.dist]newMILLER02.1954',STATUS = 'OLD')
         OPEN (UNIT =28, FILE ='[.dist]newMILLER33.1954',STATUS = 'OLD')
         OPEN (UNIT =29, FILE ='[.dist]newMILLER34.1954',STATUS = 'OLD')
         OPEN (UNIT =30, FILE ='[.dist]newMILLER35.1954',STATUS = 'OLD')
         OPEN (UNIT =31, FILE ='[.dist]newMILLER36.1954',STATUS = 'OLD')
         OPEN (UNIT =32, FILE ='[.dist]newMILLER37.1954',STATUS = 'OLD')
cc       OPEN (UNIT =33, FILE ='[.dist]newMILLER38.1954',STATUS = 'OLD')
         OPEN (UNIT =33, FILE ='[.dist]MILLER78938.1954',STATUS = 'OLD')
         OPEN (UNIT =34, FILE ='[.dist]newMILLER39.1954',STATUS = 'OLD')
         OPEN (UNIT =35, FILE ='[.dist]newMILLER40.1954',STATUS = 'OLD')
         OPEN (UNIT =36, FILE ='[.dist]newMILLER41.1954',STATUS = 'OLD')
         OPEN (UNIT =37, FILE ='[.dist]newMILLER42.1954',STATUS = 'OLD')
         OPEN (UNIT =38, FILE ='[.dist]newMILLER44.1954',STATUS = 'OLD')
         OPEN (UNIT =39, FILE ='[.dist]newMILLER45.1954',STATUS = 'OLD')
         OPEN (UNIT =40, FILE ='[.dist]newMILLER46.1954',STATUS = 'OLD')
cc       OPEN (UNIT =41, FILE ='[.dist]newMILLER47.1954',STATUS = 'OLD')
         OPEN (UNIT =41, FILE ='[.dist]MILLER78947.1954',STATUS = 'OLD')
         OPEN (UNIT= 42, FILE ='[.dist]newMILLER48.1954',STATUS = 'OLD')
C        OPEN (UNIT =43, FILE ='[.dist]newMILLER49.1954',STATUS = 'OLD')
         OPEN (UNIT =43, FILE ='[.dist]newMILLER14.1954',STATUS = 'OLD')
         OPEN (UNIT =44, FILE ='[.dist]newMILLER50.1954',STATUS = 'OLD')
         OPEN (UNIT =45, FILE ='[.dist]newMILLER51.1954',STATUS = 'OLD')
         OPEN (UNIT =46, FILE ='[.dist]newMILLER53.1954',STATUS = 'OLD')
         OPEN (UNIT =47, FILE ='[.dist]newMILLER54.1954',STATUS = 'OLD')
         OPEN (UNIT =48, FILE ='[.dist]newMILLER55.1954',STATUS = 'OLD')
         OPEN (UNIT =49, FILE ='[.dist]newMILLER56.1954',STATUS = 'OLD')
C
C       ENTER UNIT FILE NAMES FOR FIRST AND LAST STATE TO CONSIDER
C       1 TO 25 FOR TOP HALF; 26 TO 49 FOR BOTTOM HALF
C
        DO 602 IM = 26,49
        READ (IM,100) ITATE, YEAR
        READ (IM,101)MPAVG,FS,fss,css,PS,PSS,MPTEST,MCFS,MUFS,MRFS,MMS,CONS
        READ (IM,501)
        READ (IM,102)
        IJ = 0
        it(im) = itate
        NII = NCTY(IM)
        DO 603 II = 1,NII
        I = II + IJ
        READ (IM,103,ERR=604) KIPS(I) , PC(I), MCFC(I), MUFC(I),
```

```
     2  MPC(I),MMC(I), MFUC(I), EC(I), EXC(I), TMFU(I), SC(I)
        WRITE (92,103) KIPS(I) , PC(I), MCFC(I), MUFC(I),
     2  MPC(I),MMC(I), MFUC(I), EC(I), EXC(I), TMFU(I), SC(I)
       totc(1) = totc(1) + pc(i)
       totc(2) = totc(2) + mcfc(i)
       totc(3) = totc(3) + mufc(i)
       totc(4) = totc(4) + mpc(i)
       totc(5) = totc(5) + mmc(i)
       totc(6) = totc(6) + mfuc(i)
       totc(7) = totc(7) + ec(i)
       totc(8) = totc(8) + exc(i)
       totc(9) = totc(9) + tmfu(i)
       totc(10) = totc(10) + sc(i)
c
       totr(im,3) = totr(im,3) + sc(i)
c
603    CONTINUE
604    IJ = I-1
       READ (IM,104) TPC,TMCFC,TMUFC,TMPC,TMMC,TMFUC,TEC,TEXC,TTMFU,
     1TSC
       TOTS(1) = TOTS(1) + TPC
       TOTS(2) = TOTS(2) + TMCFC
       TOTS(3) = TOTS(3) + TMUFC
       TOTS(4) = TOTS(4) + TMPC
       TOTS(5) = TOTS(5) + TMMC
       TOTS(6) = TOTS(6) + TMFUC
       TOTS(7) = TOTS(7) + TEC
       TOTS(8) = TOTS(8) + TEXC
       TOTS(9) = TOTS(9) + TTMFU
       TOTS(10) = TOTS(10) + TSC
c
       totr(im,1) = tsc
c
       ACONS = TEC*(1.E6)/(2.205*365.*TPC)
       WRITE (93,114) region(itate),ITATE,TPC,TMCFC,TMUFC,TMPC,TMMC,
     1TMFUC,TEC,TEXC,TTMFU,TSC,ACONS
C
       NR = NRT(IM)
       DO 22 J = 1,NR
       READ (IM,107)
       NM = NREG(IM,J)
C
       DO 23 K=1,NM
        READ(IM,109)LL,PCL,MCFCL,MUFCL,MPCL,MMCL,MFUCL,ECL
     3,EXCL,TMFUL,SCL
C       WRITE(94,109)NM,PCL,MCFCL,MUFCL,MPCL,MMCL,MFUCL,ECL
C     3,EXCL,TMFUL,SCL
23     CONTINUE
       READ (IM,108)
        READ (IM,109)NPQ,RPCJ,RMCFCJ,RMUFCJ,RMPCJ,RMMCJ,
     1RMFUCJ,RECJ,REXCJ,RTMFUJ, RSCJ
       L = NST(im,J)
C      WRITE  (94,998) L
       RPC(L) = RPCJ
       RMCFC(L) = RMCFCJ
       RMUFC(L) = RMUFCJ
       RMPC(L) = RMPCJ
```

```
        RMMC(L) = RMMCJ
        RMFUC(L) = RMFUCJ
        REC(L) = RECJ
        REXC(L) = REXCJ
        RTMFU(L) = RTMFUJ
        RSC(L) = RSCJ

c
        totr(im,2) = totr(im,2) + rsc(l)
c
         WRITE (94,1090) reg(l),itate,L,RPC(L),RMCFC(L),RMUFC(L),RMPC(L),
      1RMMC(L),RMFUC(L),REC(L),REXC(L),RTMFU(L), RSC(L)
22      CONTINUE
        close (im)
602     CONTINUE
        DO 6020 L = 1,429
        TOT(1) = TOT(1) + RPC(L)
        TOT(2) = TOT(2) + RMCFC(L)
        TOT(3) = TOT(3) + RMUFC(L)
        TOT(4) = TOT(4) + RMPC(L)
        TOT(5) = TOT(5) + RMMC(L)
        TOT(6) = TOT(6) + RMFUC(L)
        TOT(7) = TOT(7) + REC(L)
        TOT(8) = TOT(8) + REXC(L)
        TOT(9) = TOT(9) + RTMFU(L)
        TOT(10) = TOT(10) + RSC(L)
6020    CONTINUE
        ACONSS =TOTS(7)*(1.E6)/(2.205*365.*TOTS(1))
        WRITE (93,6023) (TOTS(K),K=1,10),ACONSS
6023    FORMAT (/1X,'TOTALS',9x,08F11.0,2f10.0,F6.1)
        WRITE (94,6024) (TOT(K),K=1,10)
6022    FORMAT (/1X,'TOTALS',1X,10F12.0)
6024    FORMAT (/4X,'TOTALS',12X,10F11.0)
        WRITE (92,6022) (TOTC(K),K=1,10)
c
        do 6100 im=1,49
        totr(im,4) = totr(im,2) - totr(im,1)
        totr(im,5) = totr(im,3) - totr(im,1)
        totr(im,6) = totr(im,3) - totr(im,2)
        write (82,6101) it(im),(totr(im,k),k=1,6)
6100    continue
6101    format (i5,6f12.0)
        do 6103 im=1,49
        do 6102 k = 1,6
        tott(k) = tott(k) + totr(im,k)
6102    continue
6103    continue
        write (82,6104) (tott(k),k=1,6)
6104    format (/5x,6f12.0)
c
C       WRITE(92,102)
C       DO 605 I = 1,3071
C       WRITE (92,103)  KIPS(I) , PC(I), MCFC(I), MUFC(I),
C      2  MPC(I),MMC(I), MFUC(I), EC(I), EXC(I), TMFU(I), SC(I)
C605    CONTINUE
402     FORMAT(//)
400     FORMAT (I4,2F10.0)
```

```
401     FORMAT (3I3,I12)
  200    FORMAT (I4,I5,F6.0,F8.0,F10.0,F8.0,F7.0,F10.1,F6.1,F8.0)
  201    FORMAT (I4,4F10.0)
  300    FORMAT (3A4,I4)
c  100    FORMAT (25X,'**MILLER METHOD - MILK DISTRIBUTION**'//10X,
c     2  'STATE',I3,30X,'YEAR',I5/)
  100    FORMAT (120x,//10X,
     2  'STATE',I3,30X,'YEAR',I5/)
  101    FORMAT (10X,'AVERAGE MILK PER COW (LBS/YR) = ',F10.0/,
     2  10X,'TOTAL DAIRY FARMS = ',F10.0,'       TEST = ',F10.0/,
     9  10x,'TOTAL DAIRY COWS = ',f10.0/,
     3  10X,'TOTAL POPULATION = ',F10.0,'        TEST = ',F10.0/,
     8  10X,'TOTAL MILK PROD (Klb)- check mpc total = ',F10.0/,
     4  10X,'FARM CONSUMPTION (Klb) = ',F10.0/,
     1  10X,'USED ON FARM (Klb) = ',F10.0/,
     5  10X,'FARM RETAIL (Klb) = ',F10.0/,
     6  10X,'MANUFACTURED (Klb) = ',F10.0/,
     7  10X,'CONSUMPTION RATE (lbs per capita per year) = ',F10.2)
  501    FORMAT (///30X,'PC = POPULATION '/,
     2  30X,'MCFC = MILK CONSUMED ON FARMS'/,
     3  30X,'MUFC = MILK USED ON FARMS (non-consumption by people)'/,
     4  30X,'MPC  = TOTAL MILK PRODUCED (calculated)'/,
     5  30X,'MMC  = MILK USED FOR MANUFACTURING'/,
     6  30X,'MFUC = MILK TO SELL(does not include farm consumption)'/,
     7  30x,'EC   = EXPECTED CONSUMPTION (calculated)'/,
     8  30X,'EXC  = TEST TO SITE MANUFACTURING PLANTS'/,
     9  30X,'TMFU = TOTAL FLUID MILK CONSUMED'/,
     1  30X,'SURP = SURPLUS'//)
932     FORMAT (///30X,'PC = POPULATION '/,
     2  30X,'MCFC = MILK CONSUMED ON FARMS'/,
     3  30X,'MUFC = MILK USED ON FARMS (non-consumption by people)'/,
     4  30X,'MPC  = TOTAL MILK PRODUCED (calculated)'/,
     5  30X,'MMC  = MILK USED FOR MANUFACTURING'/,
     6  30X,'MFUC = MILK TO SELL(does not include farm consumption)'/,
     7  30x,'EC   = EXPECTED CONSUMPTION (calculated)'/,
     8  30X,'EXC  = TEST TO SITE MANUFACTURING PLANTS'/,
     9  30X,'TMFU = TOTAL FLUID MILK CONSUMED'/,30X,'SURP = SURPLUS'/,
     1  30X,'CONS = PER CAPITA CONSUMPTION RATE (g/d)'//)
  102    FORMAT (//3X,'CNTY',6X,'PC',8X,'MCFC',8X,'MUFC',8X,'MPC',8X,
     2  'MMC',10X,'MFUC',10X,'EC',9X,'EXC',8X,'TMFU',8X,'SURP'/)
1020     FORMAT(//2x,'STATE',10X,'REGION',5X,'PC',9X,'MCFC',7X,'MUFC',7X,
     2'MPC',7X,'MMC',9X,'MFUC',9X,'EC',8X,'EXC',7X,'TMFU',7X,'SURP'/)
934     FORMAT(//2x,'STATE',15X,'PC',9X,'MCFC',7X,'MUFC',7X,'MPC',7x,
     2'MMC',9X,'MFUC',8X,'EC',7X,'EXC',6X,'TMFU',6X,'SURP',4x,'CONS'/)
  103    FORMAT (I6,10F12.0)
  104   FORMAT (/,6X,10F12.0)
  114   FORMAT (1X,a,I3,8F11.0,2F10.0,F7.2)
  105    FORMAT(25I3)
  106    FORMAT (15I4)
  195    FORMAT(2I20)
  196    FORMAT (I8,15I4)
  197    FORMAT (6I10)
  107    FORMAT (/,3X,'CNTY',6X,'PC',8X,'MCFC',8X,'MUFC',8X,'MPC',8X,
     2  'MMC',10X,'MFUC',10X,'EC',9X,'EXC',8X,'TMFU',8X,'SURP'/)
  108    FORMAT(/)
  109    FORMAT (I6,10F12.0)
 1090    FORMAT (1x,a12,i3,I6,10f11.0)
```

```
  998  FORMAT (I7)
5011    FORMAT (20X,'THE VOLUME RATES OF MILK ARE IN THOUSANDS ',
     1  'OF POUNDS PER YEAR'/)
5010  format(/,20x,'[file : [.cows]milkreg3.dat] prepared 3 mar 1989',
     1  ' with newmillerus2.for',
     2  /,20x,'MILK UTILIZATION BY REGION (MILLER METHOD): YEAR 1954'/)
933    FORMAT (20X,'THE VOLUME RATES OF MILK ARE IN THOUSANDS ',
     1  'OF POUNDS PER YEAR'/,20x,'except for CONS (grams per day)')
931    format(/,20x,'[file : [.cows]milkst3.dat] prepared 3 mar 1989',
     1  ' with newmillerus2.for',
     2  /,20x,'MILK UTILIZATION BY STATE (MILLER METHOD): YEAR 1954'/)
       STOP
       END
```

ATTACHMENT A9.11 : MILKDIST.FOR

```
          program milkdist
c
c         prepared in NOVEMBER 88
c         essentially extracted from MILKUSANNIE.FOR
c         CALCULATES VOLUMES OF MILK IN EACH CATEGORY AND EACH COUNTY
c         AS WELL AS ORIGIN (OR DESTINATION) OF MILK TRANSFERRED
C
          dimension max(400),mm(400,40),reg(400),ips(56100)
          dimension surmr(400),VTOUT(400),VTIN(400),ECX(400),RATP(400)
          dimension pc(3100),vcfc(3100),vrfc(3100),vfuc(3100),ec(3100)
          dimension surpc(3100),vlm(3100),surpr(400),surp(400,35)
         dimension surm(3100),RATN(400)
          dimension pr(400),vfur(400),vlmr(400)
          dimension LJ(40)
          dimension nrg(3100),tmfu(3100)
          dimension SMR(400)
          dimension tn(400),tp(400),v2(3100),v2r(3100)
          DIMENSION V1(400),SRM(3100)
          DIMENSION smc(3100)
          dimension vtr(400,400),vin(400),vout(400)
          dimension fips(3100),dist(3100)
          DIMENSION GFMONTH(12),NAMCTY(3100),NAMST(3100)
          character*76 t1,t3,t4,t5
          character*12 reg,namcty,REGP
          character*2 NAMST
          character*15 MONTH
          CHARACTER*8 SHOT
          integer fips,GFDATE
C
c         REGUS.DAT = definition of regions
C         MILKCTY3.DAT = production and utilization of milk in each county
c                         (output of [COWS]MILLERUS)
C         BO2.DAT = county characteristics (name, FIPS code, distance from
C                   NTS, population)
C         MILKTREG.DAT = volumes of milk transferred between regions (non-zero)
C         MILKDISTR.DAT = output: volumes of milk by region
C         MILKDISTC.DAT = output: volumes of milk by county (check that the
c                          value of RER (the last one on the right) is
c                  equal or close to 0).
C
```

```
      open (unit=5, file = 'regUS.dat', status='old')
      open (unit=7, file = 'milkcty3.dat', status='old')
      open (unit=14, file = 'bO2.dat', status='OLD')
      open (unit=21, file = 'MILKtreg2.dat', status='old')
      open (unit=2, file = 'MILKDISTR.dat', status='new')
      open (unit=20, file = 'MILKDISTC.dat', status='new')
      open (unit=22, file = 'MILKPCR.dat', status='new')
c
C     CONVERSION COEFFTS FROM klb TO kg (SIF1), FROM Mlb TO kg (SIF2)
C     AND FROM days TO years (UCF)
C
      SIF1 =  1000./2.205
      SIF2 =  1.e6/2.205
      UCF  =  1./365.
C
C     READ FIPS CODES AND NAMES OF COUNTIES AND STATES
C
      DO 402 I = 1,3071
      READ (14,250) L,NAMST(I),NAMCTY(I),LIPS,DIST(I),pc(i)
      IPS(LIPS) = L
      FIPS(L) = LIPS
402   CONTINUE
C
C     DEFINITION AND ORGANIZATION OF REGIONS
C     NN = NUMBER OF REGIONS
C
      NN = 398
      do 1 n = 1,NN
      MAX(N) = 0
      ECX(N) = 0.
      VTOUT(N) = 0.
      VTIN(N) = 0.
      read (5,10) L,reg(n),Mmax
      READ (5,20) (mm(n,m),m=1,MMAX)
      MAX(N) = MMAX
      IMM = IMM + MAX(N)
      DO 1 M = 1,MMAX
      NA = MM(N,M)
      NI = IPS(NA)
      NRG(NI) = N
1     continue
C
      VLMT = 0.
      VFURT = 0.
      ECT = 0.
      SURPT = 0.
      SURMT = 0.
      V2T = 0.
      VCFT = 0.
      VRFT = 0.
      SRMT = 0.
      VTT = 0.
      SMT = 0.
      IM = 0
      nmax = 0
      ttp = 0.
      ttn = 0.
```

```
          turpr = 0.
          turmr = 0.
          tcfc = 0.
          trfc = 0.
          tlm = 0.
          t2 = 0.
          trm = 0.
          turpc = 0.
          turm = 0.
          tec = 0.
          tmc = 0.
C
C
C         MILK PRODUCTION AND DISTRIBUTION
C
C
C         REGIONAL TRANSFER IN MATRIX FORM
c         read condensed version of milk transfer between regions
c
          do 377 nd = 1,398
          do 377 mr = 1,398
          vtr(mr,nd) = 0.
377       continue
378       continue
          read (21,3770,end=379) nd,reg(nd),mr,reg(mr),vtr(mr,nd)
          go to 378
379       continue
C
          DO 3700 N = 1,398
          DO 3700 M = 1,398
          VTOUT(N) = VTOUT(N) + VTR(M,N)
          VTIN(N) = VTIN(N) - VTR(N,M)
3700      CONTINUE
          do 219 n = 1,nn
          do 219 m = 1,nn
C         CHANGE
219       vtr(n,m) = SIF1*vtr(n,m)
C         END OF CHANGE
C219      vtr(n,m) = SIF2*vtr(n,m)
C
C
C         I. ENTER DATA FROM MILLER PROGRAM
C
          IM = 3071
          do 8 i = 1,IM
        read(7,70)l,pc(i),vcfc(i),vrfc(i),r1,r2,vfuc(i),ec(i),r3,tmfu(i),
     1surpc(i)
          vlm(i) = vfuc(i)
          if(surpc(i).gt.0.) vlm(i) = ec(i)-vcfc(i)
          if (vlm(i).lt.0.) vlm(i) = 0.
8         continue
          DO 720 N=1,NN
          surpr(n) = 0.
          mmax = max(n)
          do 720 m = 1,mmax
          NA = mm(n,m)
          NI = IPS(NA)
```

```
            NRG(NI) = N
C
            ECX(N) = ECX(N) + (EC(NI)/1000.)
C
            surpr(n) = surpr(n) + surpc(ni)
720         CONTINUE
            DO 721 N = 1,NN
            SURPR(N) = SURPR(N) * SIF1
721         CONTINUE
C
            DO 3701 N = 1,398
            VTOUT(N) = VTOUT(N)/1000.
            VTIN(N) = VTIN(N)/1000.
            RATP(N) = 100.*VTOUT(N)/ECX(N)
            RATN(N) = 100.*VTIN(N)/ECX(N)
3701        CONTINUE
C
C           CONVERSION TO SI UNITS
C
            DO 722  I = 1,IM
            VCFC(I) = VCFC(I) * SIF1
            VRFC(I) = VRFC(I) * SIF1
            VFUC(I) = VFUC(I) * SIF1
            tmFU(I) = tmFU(I) * SIF1
            VLM(I) = VLM(I) * SIF1
            EC(I) = EC(I) * SIF1
            SURPC(I) = SURPC(I) * SIF1
722         CONTINUE
C
C
C           II. MILK TRANSFER BETWEEN COUNTIES OF THE SAME REGION
C
            do 3 n = 1,NN
            pr(n) = 0.
            surmr(n) = 0.
            tn(n) = 0.
            tp(n) = 0.
            SMR(N) = 0.
            mmax = max(n)
            do 2 m = 1,mmax
            nA = mm(n,m)
            NI = IPS(NA)
            surp(n,m) = surpc(ni)
            VFUR(N) = VFUR(N) + VFUC(NI)
            VLMR(N) = VLMR(N) + VLM(NI)
            pr(n) = pr(n) + pc(ni)
            v2(ni) = 0.
2           continue
3           continue
            do 4 n = 1,NN
            mmax = max(n)
            do 14 m = 1,mmax
            nA = mm(n,m)
            NI = IPS(NA)
            IF (SURPC(NI).LT.0.) TN(N) = TN(N) + SURPC(NI)
            IF (SURPC(NI).GT.0.) TP(N) = TP(N) + SURPC(NI)
            surm(ni) = surpc(ni)
```

```
14        CONTINUE
4         CONTINUE
          DO 15 N = 1,NN
          IF (TP(N).EQ.0.) GO TO 15
          IF (TN(N).EQ.0.) GO TO 15
          mmax = max(n)
          IF (SURPR(N).LT.0.) GO TO 18
          do 16 m = 1,mmax
          nA = mm(n,m)
          NI = IPS(NA)
          IF (SURPC(NI).LT.0.) GO TO 17
          SURM(NI) = SURPC(NI)*SURPR(N)/TP(N)
          GO TO 16
17        V2(NI) = -SURPC(NI)
          SURM(NI) = 0.
16        CONTINUE
          GO TO 22
18        DO 19 M = 1,MMAX
          nA = mm(n,m)
          NI = IPS(NA)
          IF (SURPC(NI).LT.0.) GO TO 21
          SURM(NI) = 0.
          GO TO 19
21        V2(NI) =  SURPC(NI)*TP(N)/TN(N)
          SURM(NI) = SURPC(NI)*SURPR(N)/TN(N)
19        CONTINUE
22        CONTINUE
15        CONTINUE
          do 5 n = 1,NN
          mmax = max(n)
          do 6 m = 1,mmax
          nA = mm(n,m)
          NI = IPS(NA)
          V2R(N) = V2R(N) + V2(NI)
          surmr(n) = surmr(n) + surm(ni)
6         CONTINUE
5         CONTINUE
C
C         III. MILK TRANSFER BETWEEN REGIONS
C
          DO 31 N = 1,NN
          V1(N) = 0.
          vout(n) = 0.
          vin(n) = 0.
31        CONTINUE
          do 214 m=1,nn
          if (surpr(m).gt.0.) go to 214
          do 215 n=1,nn
          vin(m) = vin(m) + vtr(M,N)
215       continue
          v1(m) = vin(m)
214       continue
C
C         IV. PREPARATION OF OUTPUT DATA BY COUNTY AND REGION AS WELL
C             AS FOR THE ENTIRE AREA
C
C
```

```
          DO 34 N = 1,NN
          MMAX = MAX(N)
          write (2,30) n,reg(n),mmax,tp(n),tn(n),surpr(n),SURMR(N)
          WRITE (22,3703) N,REG(N),MMAX,ECX(N),VTOUT(N),VTIN(N),
     1RATP(N),RATN(N)
          nmax = nmax + mmax
          ttp = ttp + tp(n)
          ttn = ttn + tn(n)
          turpr = turpr + surpr(n)
          turmr = turmr + surmr(n)
          DO 34 M = 1,MMAX
          nA = mm(n,m)
          NI = IPS(NA)
          IF (SURMR(N).EQ.0.) GO TO 341
          SRM(NI) = V1(N) * SURM(NI) / SURMR(N)
          GO TO 34
341       SRM(NI) = 0.
34        SMC(NI) = -EC(NI) + VCFC(NI)+VLM(NI)+V2(NI)+SRM(NI)
          DO 35 I = 1,3071
          RER = 100. * SMC(I) / EC(I)
          tcfc = tcfc + vcfc(i)
          trfc = trfc + vrfc(i)
          tlm = tlm + vlm(i)
          t2 = t2 + v2(i)
          trm = trm + srm(i)
          turpc = turpc + surpc(i)
          turm = turm + surm(i)
          tec = tec + ec(i)
          tmc = tmc + smc(i)
         write(20,120) i,fips(i),NRG(I),VCFC(I),VRFC(I),VLM(I),V2(I),
     1SRM(I),SURPC(I),SURM(I),EC(I),SMC(I),RER
35        CONTINUE
          trer = 100. * tmc / tec
          write (2,301) nmax,ttp,ttn,turpr,turmr
          write (20,121) tcfc,trfc,tlm,t2,trm,turpc,turm,tec,tmc,trer
C
3703      FORMAT (1X,I3,2X,A12,I3,5F11.3)
C
10        format (i4,1x,A12,I3)
20        format (20I6)
30        format (1x,i3,2X,a12,i3,2x,4F14.0)
70        format (i6,10F12.0)
120       FORMAT (i5,I6,i4,4F10.0,F14.0,3F12.0,F14.0,F10.3)
121       FORMAT (/,15x,4E10.3,E14.5,2F12.0,E12.5,E14.5,F10.3)
190       FORMAT (/,13X,'  CONSUMED      USED    SALES OF ','  COUNTY',
     33X,'  REGION',3X,'  CALC.',4X,'  EXPECTED   DIFF.'/13X,
     4'  ON FARM      ON FARM  LOCAL MILK',' TRANSFER ','  TRANSFER ',
     5'  TOTAL'/)
250       FORMAT (I6,1X,A2,1X,A12,I6,F10.0,f12.0)
301       format (/,6x,'TOTALS',4x,i5,2X,4F14.0)
3770      format (1x,i3,2x,a12,i3,2x,a12,f12.0)
          stop
          end
```

ATTACHMENT A9.12 : GRPDOSE1.FOR

```
      program grpdose1
c
c     revised nov 1992 from groupdose2.for
c     calculates individual doses (rads) for each of the (now) 13 age groups
c     + per capita and collective doses
c     one program per test
c
      dimension d(5,14),ud(5,14),aop(14),uaop(14),cons(9,14),dcf(14)
      dimension cop(9),ucop(9),pb(14),age(14),CM13(60),Cm14(60),MI(60)
      dimension CM9(60),Cm10(60),cm11(60),cm12(60),fmd(14),fmd2(14)
      dimension c(10),uc(10),uas(5,14),as(5,14),vx(5),vs2(5)
      dimension dm(5,14),ds2(5,14),ucons(14),ucn(14),a(5,14),ua(5,14)
      dimension apc(14),uapc(14),amk(14),uamk(14)
      character*2 namst,series
      character*12 namcty
      character*17 namser
      character*20 age
      character*8 shot
      character*80 title
      open (unit=1, file = 'newb02.dat', status = 'old')
      open (unit=2, file = 'conspop2.dat', status = 'old')
      open (unit=3, file = 'age2.dat', status = 'old')
      open (unit=4, file = 'testmilk.res', status = 'old')
      open (unit=5, file = 'testconc.res', status = 'old')
      open (unit=6, file = 'testform.dat', status = 'old')
      open (unit=11, file = 'testd2.res', status = 'new')
      open (unit=12, file = 'testd3.res', status = 'new')
      open (unit=13, file = 'testd4.res', status = 'new')
      open (unit=14, file = 'testd5.res', status = 'new')
      open (unit=15, file = 'testd6.res', status = 'new')
      open (unit=16, file = 'testd7.res', status = 'new')
      open (unit=17, file = 'testd8.res', status = 'new')
      open (unit=18, file = 'testd9.res', status = 'new')
      open (unit=19, file = 'testd10.res', status = 'new')
      open (unit=20, file = 'testd11.res', status = 'new')
      open (unit=21, file = 'testd12.res', status = 'new')
      open (unit=22, file = 'testd13.res', status = 'new')
      open (unit=23, file = 'testd14.res', status = 'new')
      open (unit=30, file = 'testpcd.res', status = 'new')
      open (unit=31, file = 'check.res', status = 'new')
c
      do 20 j=1,6
      read (2,201) title
      read (2,200) (cons(j,k),k=1,8)
      read (2,202) (cons(j,k),k=9,14)
20    continue
      read (2,201) title
      DO 27 I = 1,49
      READ (2,207) L,cm9(L),cm10(L),cm11(L),cm12(L),CM13(L),Cm14(L)
27    CONTINUE
207   FORMAT (I3,3x,f5.0,5F9.0)
      read (6,601) series,ns
      read (6,602) shot,mh,id,iy
      read (2,201) title
      read (2,203) (pb(k),k=1,8)
```

```
      read (2,204) (pb(k),k=9,14)
      read (2,201) title
      read (2,205) (dcf(k),k=1,8)
      read (2,206) (dcf(k),k=9,14)
      read (2,201) title
      read (2,203) (fmd(k),k=1,8)
      read (2,204) (fmd(k),k=9,14)
      read (2,201) title
      read (2,203) (fmd2(k),k=1,8)
      read (2,204) (fmd2(k),k=9,14)
      read (2,201) title
      read (2,203) (ucons(k),k=1,8)
      read (2,204) (ucons(k),k=9,14)
      do 51 k = 1,14
      read (3,310) age(k)
51    dcf(k) = dcf(k) * 0.001
c     uncertainties
      do 91 k = 1,14
      ucn(k) = log (ucons(k))
91    continue
c
      do 21 j = 7,8
      do 22 k = 1,4
22    cons(j,k) = 800.
      cons(j,5) = 1300.
      cons(j,6) = 1400.
      cons(j,7) = 1300.
      cons(j,8) = 1200.
      cons(j,9) = 1200.
      cons(j,10) = 1200.
      cons(j,11) = 1400.
      cons(j,12) = 1300.
      cons(j,13) = 1000.
      cons(j,14) = 800.
21    continue
      do 25 k = 1,8
25    cons(9,k) = cons(1,k)
      do 26 k = 9,14
26    cons(9,k) = 0.
c
      ucdf = log(1.8)
c
      do 71 k = 2,14
      n = k + 9
      nn = k - 1
      write (n,1437) series,ns,k,age(k),shot,mh,id,iy
71    continue
      write (30,445) series,ns,shot,mh,id,iy
c
c     read the concentrations and change the units to nCi.d/g
c
c
c     nc = 100
      nc = 3094
      read (4,831) series,ns,shot,mh,id,iy
c     write (31,831) series,ns,shot,mh,id,iy
      read (5,331) series,ns,shot,mh,id,iy
```

```
c9835     format (/////)
c9335     format (////)
          i1 = 1
          i2 = 40
          npm = 0
          cdmkt = 0.
          cdt = 0.
          do 616 ij = 1,100
          npm = npm + 1
          if (i1.ge.nc) go to 617
          if (i2.ge.nc) i2 = nc
          if (ij.ne.1) read (4,1831) series,ns
c         write (31,1831) series,ns
          if (ij.ne.1) write (30,448) series,ns
          write (30,446)
          read (4,832)
c         write (31,832)
          read (4,833)
c         write (31,833)
          if (ij.ne.1) read (5,1331) series,ns
          read (5,332)
          read (5,333)
          do 710 n = 11,23
          na = n - 9
          if (ij.ne.1) write (n,1435) series,ns,na
          if (n.gt.17) go to 72
          if (n.lt.14) go to 73
c         write (n,433)
          write (n,435)
          write (n,436)
          go to 710
72        continue
c         write (n,433)
          write (n,434)
          write (n,438)
          go to 710
73        write (n,432)
          write (n,434)
          write (n,438)
710       continue
          DO 760 I = i1,i2
          gmdmk = 0.
          gsdmk = 0.
          gmdpc = 0.
          gsdpc = 0.
          cdmk = 0.
          cd = 0.
          READ (1,829) IPS,pop
          L = IPS/1000
          cons(1,9) = cm9(L)
          cons(1,10) = cm10(L)
          cons(1,11) = cm11(L)
          cons(1,12) = cm12(L)
          cons(1,13) = cm13(L)
          cons(1,14) = cm14(L)
          read (4,830) namst,namcty,(c(k),uc(k),k=1,8)
c                write (31,830) namst,namcty,(c(k),uc(k),k=1,8)
```

```
          read (5,330) namst,namcty,(cOP(k),ucOP(k),k=1,6)
          c(6) = c(6) * 0.001
          c(7) = c(7) * 0.001
          c(8) = c(8) * 0.001
          do 52 m = 1,5
52        cop(m) = cop(m) * 0.001
c
c         calculate the intakes of I-131
c
          do 33 k = 2,14
          aop(k) = 0.
          uaop(k) = 0.
          xt = 0.
          s2t = 0.
c         if (c(1).le.0.) go to 531
          do 32 j = 2,6
          CDF = cop(j)*cons(j,k)
c         if(i.eq.1) write (31,9990) j,cdf
c9990     format (2x,'cdf',i2,e10.3)
          IF (CDF.EQ.0.) GO TO 32
          um = log (CDF)
          usig = log (ucop(j))
          usn2 = (usig * usig) + (ucn(k) * ucn(k))
          umsn2 = um + (usn2/2.)
          ux = exp(umsn2)
          us2 = ux * ux * (exp(usn2)-1.)
          xt = xt + ux
          s2t = s2t + us2
32        continue
          if (xt.le.0.) go to 33
          xt2 = xt * xt
          xx = 1. + (s2t/xt2)
          usigt = SQRT(log(xx))
          aop(k) = XT/SQRT(XX)
c         if(i.eq.1) write (31,9991) k,aop(k)
c9991     format (2x,'aop',i2,e10.3)
          uaop(k) = exp(usigt)
33        continue
531       continue
          do 34 k = 2,14
          xt = 0.
          s2t = 0.
          ux = 0.
          us2 = 0.
          do 532 kk = 1,5
          d(kk,k) = 0.
          ud(kk,k) = 0.
          dm(kk,k) = 0.
          ds2(kk,k) = 0.
          uas(kk,k) = 0.
532       continue
          if (aop(k).le.0.) go to 533
          if (uaop(k).le.0.) go to 533
          um = log(aop(k))
          usig = log(uaop(k))
          usn2 = usig * usig
          umsn2 = um + (usn2/2.)
```

```
          ux = exp(umsn2)
c         if(i.eq.1) write (31,9992) k,ux
c9992     format (2x,'ux',i2,e10.3)
          us2 = ux * ux * (exp(usn2)-1.)
533       continue
c
c         calculate the median intakes of milk (5 diets)
c
c         if (c(6).le.0.) go to 534
          as(1,k) = c(6)*cons(1,k)
          as(2,k) = c(7)*cons(7,k)
c         if (c(1).le.0.) go to 534
          as(3,k) = c(8)*cons(8,k)
          as(4,k) = 0.
          as(5,k) = cOP(1)*cons(9,k)
c         if(i.eq.1) write (31,9999) k,(as(mm,k),mm=1,5)
c9999     format (2x,'as(mm,k)',i2,5e10.3)
c       if(as(1,k).gt.0.) uas(1,k)=(log(uc(6))*log(uc(6)))+(ucn(k)*ucn(k))
       if(uc(6).gt.0.) uas(1,k)=(log(uc(6))*log(uc(6)))+(ucn(k)*ucn(k))
       if(uc(7).gt.0.) uas(2,k)=(log(uc(7))*log(uc(7)))+(ucn(k)*ucn(k))
       if(uc(8).gt.0.) uas(3,k)=(log(uc(8))*log(uc(8)))+(ucn(k)*ucn(k))
          uas(4,k) = 0.
          if(ucop(1).gt.0.) uas(5,k) = (log(ucop(1)) * log(ucop(1))) +
     1(ucn(k) * ucn(k))
c
c         calculate the mean intakes of milk (5 diets)
c
          do 535 kk = 1,5
          vx(kk) = 0.
          vs2(kk) = 0.
          if (as(kk,k).le.0.) go to 535
          vm = log(as(kk,k))
          vsn2 = uas(kk,k)
          vmsn2 = vm + (vsn2/2.)
          vx(kk) = exp(vmsn2)
c         if(i.eq.1) write (31,9993) kk,vx(kk)
c9993     format (2x,'vx(kk)',i2,e10.3)
          vs2(kk) = vx(kk) * vx(kk) * (exp(vsn2)-1.)
535       continue
c
c         median intakes of milk + foodstuffs + air (5 diets + per capita)
c
          do 536 kk = 1,5
          xt = vx(kk) + ux
          s2t = vs2(kk) + us2
          if (xt.le.0.) go to 536
          xt2 = xt * xt
          xx = 1. + (s2t/xt2)
          usigt = SQRT(log(xx))
          a(kk,k) = XT/SQRT(XX)
c         if(i.eq.1) write (31,9994) k,a(kk,k)
c9994     format (2x,'a(kk,k)',i2,e10.3)
          ua(kk,k) = exp(usigt)
536       continue
          xpc = (vx(1)*fmd(k)) + (vx(5)*fmd2(k)) + ux
          s2pc = (vs2(1)*fmd(k)*fmd(k)) + (vs2(5)*fmd2(k)*fmd2(k)) + us2
          xmk = (vx(1) * fmd(k)) + (vx(5)*fmd2(k))
```

```
          s2mk = (vs2(1)*fmd(k)*fmd(k)) + (vs2(5)*fmd2(k)*fmd2(k))
          if (xpc.le.0.) go to 534
          if (xmk.le.0.) go to 534
          xpct2 = xpc * xpc
          xxpc = 1. + (s2pc/xpct2)
          apc(k) = xpc/sqrt(xxpc)
c         if(i.eq.1) write (31,9995) k,apc(k)
c9995     format (2x,'apc(k)',i2,e10.3)
          uapc(k) = sqrt(log(xxpc))
          xmkt2 = xmk * xmk
          xxmk = 1. + (s2mk/xmkt2)
          amk(k) = xmk/sqrt(xxmk)
          uamk(k) = sqrt(log(xxmk))
534       continue
c
c         calculation of doses in rads (5 diets)
c
          do 538 kk = 1,5
          if (a(kk,k).le.0.) go to 538
          d(kk,k) = a(kk,k) * dcf(k)
          udk2 = (log(ua(kk,k)) * log(ua(kk,k))) + (udcf * udcf)
          ud(kk,k) = exp(sqrt(udk2))
          vm = log(d(kk,k))
          vsig = log(ud(kk,k))
          vsn2 = vsig * vsig
          vmsn2 = vm + (vsn2/2.)
          dm(kk,k) = exp(vmsn2)
c         if(i.eq.1) write (31,9996) k,dm(kk,k)
c9996     format (2x,'dm(kk,k)',i2,e10.3)
          ds2(kk,k) = dm(kk,k) * dm(kk,k) * (exp(vsn2) - 1.)
538       continue
539       continue
c
c         output doses for the 5 diets
c
          n = k + 9
          do 541 m = 1,5
541       if (d(m,k).le.(1.e-09)) d(m,k) = 0.
          if ((n.gt.17).or.(n.lt.14)) go to 61
          write (n,110) NAMst,NAMcty,(d(m,k),ud(m,k),m=1,5)
          go to 34
61        write (n,111) NAMst,NAMcty,(d(m,k),ud(m,k),m=1,4)
34        continue
c
c         calculation of the per capita doses
c
          xdpc = 0.
          s2dpc = 0.
          xdmk = 0.
          s2dmk = 0.
          do 790 k = 2,14
          dpc = apc(k) * dcf(k) * pb(k)
          dmk = amk(k) * dcf(k) * pb(k)
          if (dpc.le.0.) go to 790
          if (dmk.le.0.) go to 790
          dpcl = log(dpc)
          s2 = (uapc(k)*uapc(k)) + (ucdf*ucdf)
```

```
          sn2 = exp(dpcl+(0.5*s2))
          xdpc = xdpc + sn2
          s2dpc = s2dpc + (sn2 * sn2 * (exp(s2) - 1.))
          dmkl = log(dmk)
          s3 = (uamk(k)*uamk(k)) + (ucdf*ucdf)
          sn3 = exp(dmkl+(0.5*s3))
          xdmk = xdmk + sn3
          s2dmk = s2dmk + (sn3 * sn3 * (exp(s3) - 1.))
790       continue
          if (xdpc.le.0.) go to 31
          if (xdmk.le.0.) go to 31
          x2dpc = xdpc * xdpc
          xxpc = 1. + (s2dpc/x2dpc)
          gmdpc = xdpc/sqrt(xxpc)
          sidpc = sqrt(log(xxpc))
          gsdpc = exp(sidpc)
          cd = gmdpc * pop
          gsdcd = gsdpc
          x2dmk = xdmk * xdmk
          xxmk = 1. + (s2dmk/x2dmk)
          gmdmk = xdmk/sqrt(xxmk)
          sidmk = sqrt(log(xxmk))
          gsdmk = exp(sidmk)
          cdmk = gmdmk * pop
          gcdmk = gsdmk
31        continue
          cdmkt = cdmkt + cdmk
          cdt = cdt + cd
          write (30,555) namst,namcty,gmdmk,gsdmk,gmdpc,gsdpc,cdmk,cd
760       continue
          i1 = i2 + 1
          i2 = i2 + 42
          write (30,447) series,ns,npm
c         modif nov 92
c         i3 = i2 - 42
c         if (i3.ne.nc) read (4,1832) series,ns,npm
c         if (i3.ne.nc) read (5,1332) series,ns,npm
          read (4,1832) series,ns,npm
c         write (31,1832) series,ns,npm
          read (5,1332) series,ns,npm
c         end modif nov 92
          do 616 n = 11,23
          nn = n - 9
          write (n,1334) series,ns,nn,npm
616       continue
617       continue
          write (30,1143) cdmkt,cdt
1143      format (/52x,2f12.0)
442       format (3x,a2,1x,a12,2(f12.3,f7.1),1X,I12,F7.1)
445    format (1x,'TABLE SA/',A,i2,'/CD. Estimates of per capita average'
      1,' (geometric means:GM)',/,15x,'individual doses (rad) and of
      2 collective doses (man.rad),',/,15x,'and associated uncertainties
      3 (geometric standard deviations: GSD)',/,15x,'in each county of
      4 the contiguous United States resulting from',/,15x,'the test ',
      5 a,'detonated ',i2,'/',i2,'/','19',i2,'.'//1x,76('-'))
446      format (2x,'St. County',10x,'Average doses (rad) resulting from'
      1,4x,'Collective doses',/,22x,34('_'),7X,'(man.rad)',/,60x,17('_'),
```

```
     2/,20x,'milk consumption',04x,'all exposure',8x,'milk',7x,'all',/,
     343x,'routes',11x,'cons.',5x,'routes',/,
     424x,'GM',6X,'GSD',6X,'GM',6X,'GSD',9X,'GM',9X,'GM'/1x,76('-')/)
447    format(/,35X,'- ',A,i2,'/CD/',I2,' -',//,1H1)
448       format(1x,'TABLE SA/',a,i2,'/CD (continued)',/,1x,76('-'))
c
555       format (3x,a2,1x,a12,f10.3,f7.1,f10.3,f7.1,2f12.0)
830       format (3x,a,1x,a,8(1pe7.1e1,0pf5.1,2x))
831    format(1x,'TABLE ',a,'/',i2,'/M. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 (nCi d/L) a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in all categories of cows milk considered',/,18x,'for
     4each county of the contiguous United States and for the shot ',
     5A,/,18x,'detonated ',i2,'/',i2,'/','19',i2,':',/,01x,127('-'))
832       FORMAT(61x,'Originating',3x,'Originating',/,22x,'Fresh',6x,
     1'Consumed on',4x,'Retailed',5x,'from the',6x,'from another',
     24x,'Volume-',/,
     322x,'from cow',3x,'the farm',7x,'from farm',4x,'same region',3x,
     4'region',10x,'weighted',21x,'Backyard')
833       format (1x,'State County',20x,'(category 1)',2x,'(category 2)',
     12x,'(category 3)',2x,'(category 4)',4x,'average',08x,'Maximum',
     209x,'cow',/,16x,8(5x,9('-')),/,16x,8(5x,'GM    GSD'),/,
     31x,127('-'))
330       format (3x,a,1x,a,6(1pe9.2e1,0pf5.1,4x))
1334   format(/,40x,'- ',A,i2,'/D',I2,'/',i2,' -',//,1H1)
331    format(1x,'TABLE ',a,'/',i2,'/C. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in air and foodstuffs other than cows milk',/,18x,'used
     4 to calculate doses in each county',
     5' of the contiguous United States and for the shot ',A,
     6/,18x,'detonated ',i2,'/',i2,'/','19',i2,':',/,01x,122('-'))
332       FORMAT (22X,'Mothers',11x,'Goats ',
     111x,'Cottage',28x,'Leafy',13x,'Ground-level',/,22x,
     2'milk',14x,'milk',13x,'cheese',12x,'Eggs',13x,'vegetables',8x,
     3'air',/,1x,'State',1x,'County',09x,'(nCi d/L)',
     49x,'(nCi d/L)',8x,'(nCi d/kg)',8x,'(nCi d/kg)',7x,'(nCi d/kg)',
     58x,'(nCi d/m3)')
333       format (14x,6(6x,12('-')),/,14x,6(6x,'   GM    GSD'),/,
     101x,122('-'))
201       format (a80)
200       format (8f7.2)
202       format (6f7.2)
203       format (8f7.4)
204       format (6f7.4)
205       format (1x,8f5.1)
206       format (1x,6f5.1)
101       format (7x,a2,1x,a12)
310       format (a)
1435      format (1x,'TABLE SA/',A,'/',i2,'/D',i2,' (continued)',/)
1437      format (1x,'TABLE SA/',A,i2,'/D',i2,'. Estimates of average
     1(geometric means: GM) thyroid doses (rad) and associated',/,
     217x,'uncertainties (geometric standard deviations: GSD) to the' ,
     3a,' in each',/,17x,'county of the contiguous United States
     4resulting from the test ',a,'detonated ',i2,'/',i2,'/','19',i2,
     5'.',/)
432       format (50x,'Mothers diet',/,22x,70('-'))
```

```
434       format (22x,'average diet',8x,'high milk',11x,'milk from',11x,
     1'no milk',/,22x,'milk drinker',8x,'consumption',9x,'backyard cow'
     2,8x,'consumption')
435       format (22x,'average diet',8x,'high milk',11x,'milk from',11x,
     1'no milk',13x,'mothers milk',/,22x,'milk drinker',8x,'consumption'
     2,9x,'backyard cow',8x,'consumption')
436       format (22x,12('-'),8x,11('-'),9x,12('-'),08x,12('-'),08x,
     112('-'),/,
     213x,5(9x,'GM',6x,'GSD'),/)
437       format (1x,'Table SA/',A,'/',i1,'/D',i2,'. Estimates of average
     1(geometric means: GM) doses (rad) and associated uncertainties',/,
     220x,'(geometric standard deviations: GSD) to the',a,
     3'in each county of the',/,
     419x,' contiguous United States resulting from the shot ',
     5a,'detonated ',i2,'/',i2,'/','19',i2,'.',/)
438       format (22x,12('-'),8x,11('-'),9x,12('-'),08x,12('-'),/,
     113x,4(9x,'GM',6x,'GSD'),/)
439       format (1x,'Table SA/',A,'/',i2,'/D',i1,'. Estimates of average
     1(geometric means: GM) doses (rad) and associated uncertainties',/,
     220x,'(geometric standard deviations: GSD) to the',a,
     3'in each county of the',/,
     419x,' contiguous United States resulting from the shot ',
     5a,'detonated ',i2,'/',i2,'/','19',i2,'.',/)
440       format (1x,'Table SA/',A,'/',i2,'/D',i2,'. Estimates of average
     1(geometric means: GM) doses (rad) and associated uncertainties',/,
     221x,'(geometric standard deviations: GSD) to the',a,
     3'in each county of the',/,
     420x,' contiguous United States resulting from the shot ',
     5a,'detonated ',i2,'/',i2,'/','19',i2,'.',/)
400       format (a)
401       format (3x,a,1x,a,62x,3(f8.1,f4.1),f8.3,f4.1)
410       format (75x,a,10x,i2,1x,i2,3x,i2)
501       format (3x,a,1x,a,30x,4f15.1,f16.3)
601       format (8x,a,9x,i2)
602       format (6x,a,8x,3i3)
110       format (3x,a,1x,a,5(1pe10.1e1,0pf5.1,5x))
111       format (3x,a,1x,a,4(1pe10.1e1,0pf5.1,5x))
829       FORMAT (23x,i5,10x,f12.0)
835    format(1x,'TABLE ',a,'/S/M. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 (nCi d/L) a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in all categories of cows milk considered',/,18x,'for
     4each county of the contiguous United States and for the test ',
     5'test ',A,/,01x,127('-'))
1331    format(1x,'TABLE ',a,'/',i2,'/C (continued)',/,01x,122('-'))
1332    format(/,60x,'- A.',A,'/',I2,'/C.',I2,' -',//,1H1)
1831    format(1x,'TABLE ',a,'/',i2,'/M (continued)',/,1x,127('-'))
1832    format(/,60x,'- A.',A,'/',I2,'/M.',I2,' -',//,1H1)
335    format(1x,'TABLE ',a,'/S/C. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in air and foodstuffs other than cows milk',/,18x,'used
     4 to calculate doses in each county',
     5' of the contiguous United States and for the test series ',/,
     618x,A,/,01x,122('-'))
          stop
          end
```

ATTACHMENT A9.13 : PERCAP1.FOR

```
          program percap1
c
c         prepared from human5.for
c         calculates per capita doses (rads) for each county
c         one program per test
c
          dimension cons(9,14),dcf(14),pcm(60),pcd(6)
          dimension cop(9),ucop(9),pb(14),age(14),CAMK(60),CAFK(60)
          dimension c(10),uc(10),food(10)
          character*2 namst,series
          character*12 namcty
          character*10 food
          character*20 age
          character*8 shot
          character*80 title
          open (unit=1, file = 'newb02.dat', status = 'old')
          open (unit=2, file = 'dcfunc.res', status = 'old')
          open (unit=4, file = 'TESTmilk.res', status = 'old')
          open (unit=5, file = 'TESTconc.res', status = 'old')
          open (unit=6, file = 'TESTform.dat', status = 'old')
          open (unit=10, file = 'TESTpcd.res', status = 'new')
          open (unit=30, file = 'checks.res', status = 'new')
c
          read(2,810) food(1)
          do 825 mm = 1,49
          read(2,800) i,pcm(i),unc
825       continue
          do 821 i = 2,6
          if (i.ne.6) read(2,801) food(i)
          if (i.eq.6) read(2,802) food(i)
          read(2,800) mm,pcd(mm),unc
821       continue
800       FORMAT (i4,f10.6,f5.1)
810       format (1x,'** per capita dose per conc. (mrad/(nCi.d/L))',
     1' for each state for ',a,' **')
801       format (1x,'** per capita dose per unit concentration',
     1' (mrad/(nCi.d.kg) for ',a,' **')
802       format (2x,'** per capita dose per unit concentration',
     1' (rad/(nCi.d.m3) for ',a,' **')
c
          read (6,601) series,ns
          read (6,602) SHOT,MH,ID,IY
c
c
c         do 51 k = 1,14
c51       dcf(k) = dcf(k) * 0.001
c
c
          write (10,441) series,ns,shot,mh,id,iy
c
c
          nc = 3094
C         nc = 100
          read (4,831) series,ns,shot,mh,id,iy
```

```
      read (5,331) series,ns,shot,mh,id,iy
      i1 = 1
      i2 = 40
      npm = 0
      do 616 ij = 1,100
      npm = npm + 1
      if (i1.ge.nc) go to 617
      if (i2.ge.nc) i2 = nc
      if (ij.ne.1) read (4,1831) series,ns
      read (4,832)
      read (4,833)
      if (ij.ne.1) read (5,1331) series,ns
      read (5,332)
      read (5,333)
      DO 760 I = i1,i2
      READ (1,1829) IPS,pop
      l = IPS/1000
      read (4,830) namst,namcty,(c(k),uc(k),k=1,8)
      read (5,330) namst,namcty,(cOP(k),ucOP(k),k=1,6)
      if (i.eq.68) write(30,830) namst,namcty,(c(k),uc(k),k=1,8)
      if (i.eq.68) write(30,330) namst,namcty,(cOP(k),ucOP(k),k=1,6)
      c(6) = c(6) * 0.001
      c(7) = c(7) * 0.001
      c(8) = c(8) * 0.001
      do 52 m = 1,5
52    cop(m) = cop(m) * 0.001
c
c     calculate the doses in rads
c
c     do 33 k = 2,14
      dop = 0.
      udop = 0.
      xt = 0.
      s2t = 0.
      if (c(1).le.0.) go to 531
      do 32 j = 2,6
      CDF = cop(j)*pcd(j)
      IF (CDF.EQ.0.) GO TO 32
      um = log (CDF)
      usig = log (ucop(j))
      usn2 = usig * usig
      umsn2 = um + (usn2/2.)
      ux = exp(umsn2)
      us2 = ux * ux * (exp(usn2)-1.)
      xt = xt + ux
      s2t = s2t + us2
      if (i.eq.68) write(30,603) ips,l,k,j,ux,us2
603   format (4i6,2e10.3)
32    continue
      xt2 = xt * xt
      xx = 1. + (s2t/xt2)
      usigt = SQRT(log(xx))
      dop = XT/SQRT(XX)
      udop = exp(usigt)
      if (i.eq.68) write(30,604) k,j,dop,udop
604   format (2i6,2e10.3)
c33   continue
```

```
531       continue
c         do 34 k = 2,14
          xt = 0.
          s2t = 0.
          ux = 0.
          us2 = 0.
          dpcm = 0.
          dpc = 0.
          udpcm = 0.
          udpc = 0.
          if (c(1).le.0.) go to 533
          um = log(dop)
          usig = log(udop)
          usn2 = usig * usig
          umsn2 = um + (usn2/2.)
          ux = exp(umsn2)
          us2 = ux * ux * (exp(usn2)-1.)
533       continue
          dpcm = c(6) * pcm(l)
          udpcm = uc(6)
          vx = 0.
          vs2 = 0.
          if (dpcm.le.0.) go to 535
          vm = log(dpcm)
          vsig = log(udpcm)
          vsn2 = vsig * vsig
          vmsn2 = vm + (vsn2/2.)
          vx = exp(vmsn2)
          vs2 = vx * vx * (exp(vsn2)-1.)
          if (i.eq.68) write(30,605) k,kk,vx,vs2
605       format (10x,2i6,2e10.3)
535       continue
          xt = vx + ux
          s2t = vs2 + us2
          if (xt.le.0.) go to 536
          xt2 = xt * xt
          xx = 1. + (s2t/xt2)
          usigt = SQRT(log(xx))
          dpc = XT/SQRT(XX)
          udpc = exp(usigt)
536       continue
          cd = dpc * pop
          ucd = udpc
          write (10,442) ips,dpcm,udpcm,dpc,udpc,cd,ucd
534       continue
c34       continue
31        continue
760       continue
          i1 = i2 + 1
          i2 = i2 + 42
          read (4,1832) series,ns,npm
          read (5,1332) series,ns,npm
616       continue
617       continue
c
442       format (6x,i6,4x,3(f12.3,f7.3))
830       format (3x,a,1x,a,8(1pe7.1e1,0pf5.1,2x))
```

```
1832   format(/,60x,'- A.',A,'/',I2,'/M.',I2,' -',//,1H1)
1831   format(1x,'TABLE ',a,'/',i2,'/M (continued)',/,1x,127('-'))
831   format(1x,'TABLE ',a,'/',i2,'/M. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 (nCi d/L) a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in all categories of cows milk considered',/,18x,'for
     4each county of the contiguous United States and for the shot ',
     5A,/,18x,'detonated ',i2,'/',i2,'/','19',i2,':',/,01x,127('-'))
832       FORMAT(61x,'Originating',3x,'Originating',/,22x,'Fresh',6x,
     1'Consumed on',4x,'Retailed',5x,'from the',6x,'from another',
     24x,'Volume-',/,
     322x,'from cow',3x,'the farm',7x,'from farm',4x,'same region',3x,
     4'region',10x,'weighted',21x,'Backyard')
833       format (1x,'State County',20x,'(category 1)',2x,'(category 2)',
     12x,'(category 3)',2x,'(category 4)',4x,'average',08x,'Maximum',
     209x,'cow',/,16x,8(5x,9('-')),/,16x,8(5x,'GM    GSD'),/,
     31x,127('-'))
330       format (3x,a,1x,a,6(1pe9.2e1,0pf5.1,4x))
1331   format(1x,'TABLE ',a,'/',i2,'/C (continued)',/,01x,122('-'))
1332   format(/,60x,'- A.',A,'/',I2,'/C.',I2,' -',//,1H1)
1333   format(/,40x,'- SA.',A,'/',I2,'/D',I2,'/',i2,' -',//,1H1)
331   format(1x,'TABLE ',a,'/',i2,'/C. Estimates of average (geometr
     1ic means: GM) time-integrated concentrations of I-131 a
     2nd associated',/,18x,'uncertainties (geometric standard deviati
     3ons: GSD) in air and foodstuffs other than cows milk',/,18x,'used
     4 to calculate doses in each county',
     5' of the contiguous United States and for the shot ',A,
     6/,18x,'detonated ',i2,'/',i2,'/','19',i2,':',/,01x,122('-'))
332       FORMAT (22X,'Mothers',11x,'Goats ',
     111x,'Cottage',28x,'Leafy',13x,'Ground-level',/,22x,
     2'milk',14x,'milk',13x,'cheese',12x,'Eggs',13x,'vegetables',8x,
     3'air',/,1x,'State',1x,'County',09x,'(nCi d/L)',
     49x,'(nCi d/L)',8x,'(nCi d/kg)',8x,'(nCi d/kg)',7x,'(nCi d/kg)',
     58x,'(nCi d/m3)')
333       format (14x,6(6x,12('-')),/,14x,6(6x,'   GM    GSD'),/,
     101x,122('-'))
201       format (a80)
200       format (8f7.2)
202       format (6f7.2)
203       format (8f7.4)
204       format (6f7.4)
205       format (1x,8f5.1)
206       format (1x,6f5.1)
101       format (7x,a2,1x,a12)
310       format (a)
431       format (1x,'Table SA/',A,'/',i1,'/D',i1,'. Estimates of average
     1(geometric means: GM) doses (rad) and associated uncertainties',/,
     219x,'(geometric standard deviations: GSD) to the',a,
     3'in each county of the',/,
     419x,'contiguous United States resulting from the shot ',
     5a,'detonated ',i2,'/',i2,'/','19',i2,'.',/)
432       format (50x,'Mothers diet',/,22x,70('-'))
434       format (22x,'average diet',8x,'high milk',11x,'milk from',11x,
     1'no milk',/,42x,'consumption',9x,'backyard cow',8x,'consumption')
435       format (22x,'average diet',8x,'high milk',11x,'milk from',11x,
     1'no milk',13x,'mothers milk',/,42x,'consumption',9x,'backyard ',
     2' cow',8x,'consumption')
```

```
436       format (22x,12('-'),8x,11('-'),9x,12('-'),08x,12('-'),08x,
     112('-'),/,
     213x,5(9x,'GM',6x,'GSD'),/)
437       format (1x,'Table SA/',A,'/',i1,'/D',i2,'. Estimates of average
     1(geometric means: GM) doses (rad) and associated uncertainties',/,
     220x,'(geometric standard deviations: GSD) to the',a,
     3'in each county of the',/,
     419x,' contiguous United States resulting from the shot ',
     5a,'detonated ',i2,'/',i2,'/','19',i2,'.',/)
438       format (22x,12('-'),8x,11('-'),9x,12('-'),08x,12('-'),/,
     113x,4(9x,'GM',6x,'GSD'),/)
439       format (1x,'Table SA/',A,'/',i2,'/D',i1,'. Estimates of average
     1(geometric means: GM) doses (rad) and associated uncertainties',/,
     220x,'(geometric standard deviations: GSD) to the',a,
     3'in each county of the',/,
     419x,' contiguous United States resulting from the shot ',
     5a,'detonated ',i2,'/',i2,'/','19',i2,'.',/)
441    format (1x,'Table SA/',A,'/',i2,'. Estimates of per capita average
     1 (geometric means:GM) doses (rad) and associated uncertainties',/,
     221x,'(geometric standard deviations: GSD) in each county of the',
     3/,20x,' contiguous United States resulting from the shot ',
     4a,'detonated ',i2,'/',i2,'/','19',i2,'.',//,8x,'fips',8x,'milk',
     52x,'unc',9x,'all',5x,'unc',5x,'man.rads'/)
400       format (a)
401       format (3x,a,1x,a,62x,3(f8.1,f4.1),f8.3,f4.1)
410       format (75x,a,10x,i2,1x,i2,3x,i2)
501       format (3x,a,1x,a,30x,4f15.1,f16.3)
601       format (8x,a,9x,i2)
602       format (6x,a,8x,3i3)
110       format (3x,a,1x,a,5(1pe10.1e1,0pf5.1,5x))
111       format (3x,a,1x,a,4(1pe10.1e1,0pf5.1,5x))
1829      FORMAT (23x,i5,12x,f10.0)
          stop
          end
```