An MS-DOS-Based Program for Analyzing Plutonium Gamma-Ray Spectra

Volume 3 Software Listings

W. D. Ruhter

W. M. Buckley

Manuscript date: September 7, 1989

LAWRENCE LIVERMORE NATIONAL LABORATORY
University of California • Livermore, California • 94551

Available to DOE and DOE contractors from the Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831 Prices available from (615) 576-8601, FTS 626-8401

Available to the public from the National Technical Information Service, U.S. Department of Commerce
5285 Port Royal Rd., Springfield, VA 22161 • A04 • (Microfiche A01)

MASTER
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Contents

Abstract.....	1
File: IA.FOR.....	3
File: CICERO.FOR.....	19
File: ANRESULT.FOR.....	22
File: AVE.FOR.....	25
File: BKGD.FOR.....	26
File: BKGRD.FOR.....	28
File: BWF.FOR.....	30
File: C2JDAT.FOR.....	31
File: DCHECK.FOR.....	32
File: CTAIL.FOR.....	35
File: DCHDAT.FOR.....	37
File: DECAY.FOR.....	38
File: DIAEA.FOR.....	39
File: DSAVE.FOR.....	40
File: GFIT2.FOR.....	42
File: GETD.FOR.....	44
File: IDPMY.FOR.....	46
File: MASTER.FOR.....	47
File: MATINV.FOR.....	48
File: MAXVAL.FOR.....	50
File: MORE.FOR.....	51
File: NCNTS.FOR.....	52
File: PUFITY.FOR.....	53
File: RDBIN.FOR.....	55
File: RDSK.FOR.....	56
File: SPOW.FOR.....	57
File: STRIP1.FOR.....	59
File: STRLEN.FOR.....	60
File: TESTBIN.FOR.....	61
File: TTEST.FOR.....	62
File: WRTEMP.FOR.....	63
File: IAEA.CMN.....	64
File: TOP.CMN.....	65
File: IAEA.PRG.....	66
File: GETIADAT.PRG.....	70
File: DECNUM.PRG.....	82
File: INPUTBAN.PRG.....	83
File: GETCOOL.PRG.....	84
File: ENDITALL.PRG.....	85
File: GETKEY.PRG.....	86
File: DCH.PRG.....	87
File: CHECKIA.PRG.....	89
File: CICDATA.PRG.....	91
File: FUNKEY.PRG.....	105
File: CHDEFALT.PRG.....	106
File: GETTIME.PRG.....	111
File: GETNUM.PRG.....	114
File: HELP.PRG.....	117
File: GETDATE.PRG.....	118
File: GETSTRG.PRG.....	122

File: GETWHNUM.PRG	124
File: IA.BAT.....	126
File: FOR.MAK	127
File: PRG.MAK.....	128

An MS-DOS-Based Program for Analyzing Plutonium Gamma-Ray Spectra

Volume 3 Software Listings

Abstract

A plutonium gamma-ray analysis system that operates on MS-DOS-based computers has been developed for the International Atomic Energy Agency (IAEA) to perform in-field analysis of plutonium gamma-ray spectra for plutonium isotopes. The program titled IAEAPU consists of three separate applications: (1) a data-transfer application for transferring spectral data from a CICERO multichannel analyzer to a binary data file, (2) a data-analysis application to analyze plutonium gamma-ray spectra for plutonium isotopic ratios and weight percents of total plutonium, and (3) a data-quality assurance application to check spectral data for proper data-acquisition setup and performance. Volume 3 contains the software listings for these applications.

```

C
C $Large
C Subroutine IAEAPU
C *****
C
C IAEAPU
C =====
C
C *****
C
C Developed by : Safeguards Technology Program
C               Nuclear Chemistry Division
C               Lawrence Livermore National Laboratory
C               Livermore, CA. 94550
C
C This work supported by LLNL PQ# 9107605
C *****
C
C Title       : IAEAPU
C Revision    : 4.2
C Author      : Wayne D. Ruhter
C               Revised for MS-DOS machines by Russell Fleming
C               COMPUTECH, Inc.
C               30998 Huntwood Avenue, Suite 107
C               Hayward, CA. 94544
C               (415)489-6514
C
C Date       : 2/2/88
C Language   : Microsoft Fortran Ver. 4.01
C *****
C
C Abstract :
C This program analyzes plutonium gamma-ray spectra for plutonium
C isotopic ratios and weight percents of total plutonium. The
C program is also capable of analyzing gamma-ray spectra of mixed
C uranium and plutonium samples for plutonium isotopes and U-235
C to total plutonium.
C The program analyzes gamma-ray peaks beginning at 123-keV up to
C 383-keV. For a complete analysis to occur the spectrum must start
C at or below 121.5 keV and include the 208.0-keV U-237 peak. The
C only input required to the program is the positions of the Pu-241
C 148-keV and the U-237 208-keV peaks. This version is an adaptation
C of the program written for the IAEA to interact with the Silene
C CICERO MCA. This version will run on an MS-DOS based system
C and read the data from a disk file.
C *****
C
C Arguments :
C *****
C
C Routines
C Called   : CTAII
C           BWF
C           BKGD
C           BKGRD
C           RDSK
C           MATINV
C           GFIT2
C           MAXVAL
C           STRIP1
C           NCNVS

```

```

C
C WRTMP
C PUFITY
C IDPWY
C AVE
C MORE
C
C Libraries
C Used :
C
C *****
C Revision History ( Include date, revision I.D., and details ) *
C *****
C
C Date: 1-NOV-80
C
C Update: 20-OCT-86 VERSION 2.6
C Modified weight calculations for ratio determination
C and modified 332- and 345- keV area calculations
C
C Update 9-OCT-86
C Modified relative efficiency, removed 148 peak
C
C Update 11-OCT-86
C Added Covariant terms to error analysis for
C isotopic abundances
C
C Update 22-JUN-87
C Modified to run as a subprogram within a larger
C executable file
C
C Update 15-AUG-87
C Modified to print out peak areas only when option selected
C
C Update 2-NOV-87
C Put in decay correction and dropped channel analysis
C
C Update 12-JUL-88 Version 3.5
C Changed Pu-241 160-keV branching ratios and modified
C plutonium weight percent abundance errors. Corrected
C 345-keV area error calculation.
C *****
C COMMON and variable declarations
C *****
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C $INCLUDE: 'iaea.cmn'
C *****
C
C COMMON /INPUT/ nref,tp239,tp239,tp240,tp241,tp242,ta241,
C x tpe239,tpe239,tpe240,tpe241,tpe242,tae241,
C x coltime,fname,rname,bin,printer,extflag,rep,
C x spectime,spectdate,Amdate,Pdate,sample
C
C CHARACTER BIN*1,PRINTER*1,EXTFLAG*1,RNAME*20,FNAME*30,JUNK*1
C character rep*1,spe,date*10,spectime*10,amdate*10,pdate*10
C character sample*10
C
C INTEGER*4 Nref(2)
C *****
C
C $INCLUDE: 'laeeds.cmn'
C *****
C
C common /dbsave/ cp238,

```



```

6 (PCT(6),A241),
1 (PCTE(1),PE238),
2 (PCTE(2),PE239),
3 (PCTE(3),PE240),
4 (PCTE(4),PE241),
5 (PCTE(5),PE242),
6 (PCTE(6),AE241)

DOUBLE PRECISION SUM,ALPHA

```

```

C THE BYTE STRING BELOW CONTAINS INFORMATION NEEDED IN FITTING
C EACH PEAK OR PEAK GROUPING. THE FIRST VALUE ON EACH LINE
C INDICATES THE NUMBER OF PEAKS IN THE GROUP. FOR INSTANCE
C THE FIRST 6 ON THE FIRST LINE INDICATES THERE ARE 6 PEAKS
C IN THE FIRST GROUP. THE REMAINING VALUES IN THE LINE INDICATE
C WHETHER THE PEAK HEIGHT IS A FREE PARAMETER FOR THAT PEAK
C OR IF IT IS FIXED RELATIVE TO ANOTHER PEAK. IF THE VALUE IS 0
C OR IS EQUAL TO THE NUMBER OF PEAKS IN THE GROUP, THE PEAK
C HEIGHT IS A FREE PARAMETER. ANY OTHER VALUE INDICATES THAT
C PEAK IS FIXED RELATIVE TO THE FREE PEAK INDICATED BY THAT VALUE.
C FOR INSTANCE, IN THE FIRST LINE THE VALUE 3 FOR THE SECOND,THIRD,
C AND FOURTH PEAK INDICATES THAT THESE PEAKS ARE FIXED RELATIVE
C TO THE THIRD FREE PEAK IN THE GROUP WHICH IN THIS CASE IS
C THE LAST PEAK. THE PEAK WHICH HAS A VALUE EQUAL TO THE NUMBER
C OF PEAKS IN THE GROUP IS THE PEAK TO WHICH THE POSITIONS OF
C ALL THE OTHER PEAKS ARE FIXED TO, i.e., IT IS THE POSITION
C REFERENCE FOR ALL OF THE OTHER PEAKS IN THE GROUP.

```

```

2 DATA STR/'6','0','3','3','3','0','1','6',
3 '1','1','1',
4 '7','0','3','0','0','0','0','7','0',
5 '1','1',
6 '1','1',
7 '1','1',
8 '8','0','1','6','0','6','0','0','0','8',
9 '7','4','4','0','0','0','0','7','0','1',/

```

```

C OUTPUT DEVICE LOGICAL UNIT #

```

```
DATA NOEV/0/

```

```

C INPUT DEVICE LOGICAL UNIT #

```

```
DATA INDEV /2/

```

```

c Line number counter for interrupted output to screen
i1=100

```

```

DATA ISOMM/238,239,240,241,242/
data laren/129,165,203,208,267,345,375/

```

```

c half lives of isotopes in minutes

```

```

data th1/4.613e07,1.2886e10,3.44e09,7.574e06,
+ 1.979e11,2.277e08/

```

```

ENERGIES (KEV) OF PEAKS TO BE ANALYZED

```

```

C DATA ENG/122.994,123.62,124.51,125.21,
1 125.209,129.294,141.567,143.35,
2 143.77,144.211,148.077,148.597,
3 148.567,150.11,152.68,154.2,
4 159.961,160.00,160.313,161.479,
5 163.35,164.597,165.926,185.72,
6 203.537,208.,202.1,205.33,267.54,
7 332.390,332.857,335.430,336.104,
8 337.760,340.486,341.517,345.014,
9 367.050,368.546,368.603,370.933,
1 372.47,375.039,376.595/

```

```

C START POSITIONS IN KEV FOR EACH PEAK OR PEAK GROUPING
C DATA ENSTCH/120.,140.3,151.8,157.4,
5 184.0,201.4,206.1,266.4,330.5,366./
C STOP POSITIONS IN KEV FOR EACH PEAK OR PEAK GROUPING
C DATA ENECH/130.6,150.9,153.8,166.8,
5 187.,205.,211.7,268.7,347.14,377.9/
C NUMBER OF CHANNELS USED TO DETERMINE THE BACKGROUND IN
C IN FRONT OF EACH GROUP
C DATA NBG1/2,10,4,5,5,3,5,5,20,8/
C NUMBER OF CHANNELS USED TO DETERMINE THE BACKGROUND
C BEHIND EACH GROUP
C DATA NBG2/10,3,3,7,5,3,10,10,10,3/
C OFFSET ENERGIES FOR EACH GROUP TO SET STARTING CHANNEL FOR
C SEARCHING FOR REFERENCE PEAK
C DATA SE/5.3,7.4,0.,6.1,0.,0.,0.,11.5,7.0/
C INTENSITY RATIOS FOR INDICATED PEAKS

```

```

DATA RATIO/
1 0.00315,
2 0.00993,
3 0.009,
4 0.42,
5 0.1605,
6 0.05,
7 0.9283,
8 0.203,
9 0.055095,
1 0.057515,
2 0.19426,
3 0.16496,
4 1.,
5 /

```

```

C ENERGIES OF REFERENCE PEAKS USED TO ESTABLISH GAIN, ZERO, AND
C PEAK SHAPE PARAMETERS

```

```

C FU-242 VALUE ENTERED FLAG
C P242FG = .TRUE.

```

```

C STD(1)=148.567
1 STD(2)=208.
C Reset, could have been changed by U analysis
ENSTCH(6)=201.4
ENECH(6) #205.
NBG2(6) #2

```

```

C ---[ Set Debug Flag ]---

```

```

C DEBUG=.FALSE.

```

```

C *****
C format declarations
C *****

```

```

C
139 format(/, ' Report file ', A, /, ' exists. Overwrite it ? ', $)
140 format(/, ' There is a problem with creating ', A)
141 format(/, ' Aborting analysis, retry with a different report name')
200 FORMAT(/, 1H0, 113, ' IAEA PU ANALYSIS OF SPECTRUM:', A)
201 format(/, 113, ' Pu238/Pu239 = ', f11.6, ' + ', f11.6, /,
+      113, ' Pu240/Pu239 = ', f11.6, ' + ', f11.6, /,
+      113, ' Pu241/Pu239 = ', f11.6, ' + ', f11.6, /,
+      113, ' Am241/Pu241 = ', f11.6, ' + ', f11.6, /,
+      113, ' U235/Pu239 = ', f11.6, ' + ', f11.6, /)
202 FORMAT(1X, 128, ' PLUTONIUM AGE = ', F9.2, ' + ', F9.2, ' DAYS', /, 1X,
1 142, ' = ', F9.2, ' + ', F9.2, ' YEARS', /, 1X)
203 FORMAT(/, 124, ' WT.PCT. ABS. ERR. /,
211 FORMAT(/, 125, ' MEAS. DECAT CORR. ')
213 FORMAT(124, ' WT.PCT. ABS. ERR. OPERATOR WT.PCT. (M - O)%
1')
214 FORMAT(124, '
204 FORMAT(5(/, 125, ' PUT', 13, ' = ', F7.4, ' + ', F6.4, /, 1X)
205 FORMAT(117, ' AM241 = ', F7.4, ' + ', F6.4, /,
1 //, 19, ' U235/PU RATIO = ', F7.4, ' + ', F6.4, /,
206 FORMAT(118, ' AM241 = ', F7.4, ' + ', F6.4, 10X, F7.4, 11X, F7.4, /,
1 110, ' U235/PU RATIO = ', F7.4, ' + ', F6.4, /,
207 FORMAT(/, 17, ' PU240 EFFECTIVE = ', F7.4, ' + ', F6.4, /,
209 FORMAT(/, ' SPECIFIC POWER (mW/g) = ', F7.4, ' + ', F6.4, /,
210 FORMAT(' Declared isotopic values not entered. I**2 test'
1 ' NOT performed.')
212 FORMAT(19, ' VALUE OF I**2 = ', F13.3, /)
300 FORMAT(117, ' PU', 13, ' = ', F7.4, ' + ', F6.4, /,
301 FORMAT(117, ' PU', 13, ' = ', F7.4, ' + ', F6.4, 3X, A, /,
305 FORMAT(118, ' PU', 13, ' = ', F7.4, ' + ', F6.4, 10X, F7.4, 11X, F7.4, /,
306 FORMAT(118, ' PU', 13, ' = ', F7.4, ' + ', F6.4, 3X, A, /,
400 format(' Sample Name: ', A10, /)
800 format(1H1, /)
801 FORMAT(14)
802 FORMAT(12)
903 FORMAT(06.3)
901 FORMAT(A)
905 FORMAT(' THERE IS A PROBLEM WITH SPECTRUM NAMED: ', A)
909 FORMAT(' IAEA PU VERSION 4.21, /, ANALYSIS OF: ', A)
910 FORMAT(' GAIN = ', F7.4, /, ' ZERO = ', F8.3, /, ' FWHM(208 KEV) = ', F5.3)
911 format(' Spectrum Date: ', A, /, ' Spectrum Time: ', A,
+      /, ' Spectrum ID #: ', F5.0,
+      /, ' Spectrum Livetime: ', F8.0, ' seconds')
912 format(' Spectrum total counts: ', F10.0, /)
913 FORMAT(' Oper. Pu Analysis date: ', A)
914 FORMAT(' Oper. Am Analysis date: ', A, /)
920 FORMAT(' PEAK EN. COUNTS XER')
921 FORMAT(/, 1X,
+      ' IAEA Plutonium Analysis Diagnostics ----', /)
922 FORMAT(/, ' GROUP #', 12)
925 FORMAT(' REDUCED CHI-SQUARE = ', F6.1)
940 FORMAT(F7.0, 3F11.2)
970 format(1X, 13, ' -kev peak area is <= 0. Analysis aborted!')
9101 format(' 148-kev peak position entered: ', I6)
9102 format(' 208-kev peak position entered: ', I6)
9400 FORMAT(' SHAP(1) = ', F8.5, /, ' SHAPC(2) = ', F8.5, /, ' ATAIL = ', F5.3,
+      /, ' BTAIL = ', F5.3)
9500 format(' Working on group ', 12)
C
C .....
C
5 call c1s
C
C ---[ Get info from OUT_1A.PRM ] ---

```

```

C
call getd
if (exflag.eq.'X') goto 999
if ((tp238.lt.0).or.(tp239.lt.0).or.(tp240.lt.0).or.
+ (tp241.lt.0).or.(tp242.lt.0).or.(ta241.lt.0)) then
tp238=0
tp239=0
tp240=0
tp241=0
tp242=0
ta241=0
endif
p238= tp238
p239= tp239
p240= tp240
p241= tp241
p242= tp242
a241= ta241
pe238= tpe238
pe239= tpe239
pe240= tpe240
pe241= tpe241
pe242= tpe242
ae241= tae241
wpct(1)=p238
wpct(2)=p239
wpct(3)=p240
wpct(4)=p241
wpct(5)=p242
wpct(6)=a241
idev=6
call testbin(fname, bin, idev)
IF (BIN.EQ.'B') THEN
OPEN(INDEV, FILE=FNAME, STATUS='OLD', FORM='BINARY',
+ IOSTAT=TEMP)
ELSE
OPEN (INDEV, FILE=FNAME, STATUS='OLD', ACCESS='SEQUENTIAL',
+ IOSTAT=TEMP)
ENDIF
IF (TEMP.NE.0) GOTO 6
GO TO 7
6 WRITE(*, 905) FNAME
write(*, *) 'Press Enter to Continue'
read(*, 601) jnk
GO TO 999
C
C READ DATA FILE INTO DATA ARRAY
C
7 tempnum=enech(1)
C
IF (BIN.EQ.'B') THEN
CALL RDBIN(NUMCHS, numDATA, INDEV)
do 1234 i=1, numchs
data(i)=dble(numdata(i))
1234 continue
ELSE
CALL RDSK(NUMCHS, DATA, INDEV)
ENDIF
enech(1)=tempnum
C
C .....
C ---[ Dropped channel analysis ] ---
C

```



```

do 1235 i=3,nums
  if ((i.gt.3).and.(data(i-1).ge.20).and.(data(i+1).ge.20)
    .and.(data(i).eq.0))then
    write(*,*)
    write(*,*)'Zero count channel found at channel number',i
    write(*,*)'Aborting analysis. Transfer data again and retry'
    write(*,*)'analysis with new spectral data.'
    write(*,*)
    write(*,*)'Press Enter to Continue'
    read(*,901)jnk
    goto 999
  endif
1235 continue

C --- Printer = 0,S,P
  IF (PRINTER.EQ.'P') THEN
    write(*,*)
    write(*,*)'Report going to the printer, make sure it is on-line'
    write(*,*)
    write(*,*)'Press Enter to Continue'
    read(*,901)jnk
    write(*,*)'Ok'
    OPEN(2,FILE='PRN')
    NDEV=2
  elseif (printer.eq.'D') then
    write(*,*)
    write(*,*)'Report going to disk.'
    write(*,*)'Make sure drive is ready and there is free disk space'
    write(*,*)
    write(*,*)'Press Enter to Continue'
    read(*,901)jnk
    write(*,*)'Ok'

    inquire(FILE=rname,EXIST=exists)
    if (exists) then
      write(*,139)rname
      read(*,901)jnk
      if ((jnk.eq.'N').or.(jnk.eq.'n')) then
        write(*,141)
        write(*,*)'Press Enter to continue'
        read(*,901)jnk
        goto 999
      else
        open(2,FILE=RNAME,status='OLD',access='SEQUENTIAL',
          10STAT=fstat)
        if (fstat.gt.0) then
          write(*,140)rname
          write(*,*)'Press Enter to Continue'
          read(*,901)jnk
          goto 999
        endif
      endif
    else
      open(2,FILE=RNAME,status='NEW',access='SEQUENTIAL',
        10STAT=fstat)
      if (fstat.gt.0) then
        write(*,140)rname
        write(*,*)'Press Enter to Continue'
        read(*,901)jnk
        goto 999
      endif
    endif
  endif
  ndev=2
else
  ndev=0
endif

```

```

  call cls
  call locate(2,1,ierr)
ENDIF

  if (debug) then
    write(*,*)' Samplenum="'sample,'"
    write(*,*)'1234567890123456789012345678901234567890'
  endif

C --- [ If sample flag is true, write results to Anresult.dbf ] ---

  if (sample.eq.' ') then
    samflag=.false.
  else
    samflag=.true.
  endif
  if (debug) write(*,*)'Sample flag=',samflag,
    * ' sample="'sample,'"

C
C -----
C --- [ Decay correction ] ---
C -----
C
  dopudecay=.false.
  doamdecay=.false.
  nodecay=.false.

C --- convert specdate,Pudate and Amdate to integers for month, day and year

C --- Specdate's month,day,year
  syspecdate(1:2)
  sm=specdate(4:5)
  sd=specdate(7:8)
  if ((sm.eq.' ').or.(sd.eq.' ').or.(sy.eq.' '))then
    write(ndev,*)
    write(ndev,*)' Spectrum acquisition date is not defined.'
    write(ndev,*)' decay correction will not be performed.'
    write(ndev,*)
    nodecay=.true.
    goto 555
  endif
  read(sm,802)smon
  read(sd,802)sday
  read(sy,802)syear
  if (debug) write(*,*)' spec month=',sm,' month=',smon
  if (debug) write(*,*)' spec day=',sd,' day=',sday
  if (debug) write(*,*)' spec year=',sy,' year=',syear

C --- Pudate's month,day,year
  py=pudate(1:2)
  pm=pudate(4:5)
  pd=pudate(7:8)
  if ((pm.eq.' ').or.(pd.eq.' ').or.(py.eq.' '))then
    write(ndev,*)
    write(ndev,*)' Plutonium analysis date is not defined, Plutonium'
    write(ndev,*)' decay correction will not be performed.'
    write(ndev,*)
    dopudecay=.false.
    goto 1222
  endif
  read(pm,802)pmom
  read(pd,802)pdlay
  read(py,802)pyear
  dopudecay=.true.
  if (debug) write(*,*)' Pu month=',pm,' month=',pmom
  if (debug) write(*,*)' Pu day=',pd,' day=',pdlay

```

```

      if (debug) write(*,*) ' Pu year=',py,' year=',pyear
c      Amdate's month,day,year
1222  ay=amdate(1:2)
      am=amdate(4:5)
      ad=amdate(7:8)
      if ((am.eq.' ') .or. (ad.eq.' ') .or. (ay.eq.' ')) then
        write(ndev,*)
        write(ndev,*) ' Americium analysis date is not defined, Americium'
        write(ndev,*) ' decay correction will not be performed.'
        doamdecays=.false.
        goto 554
      endif
      read(am,802)amon
      read(ad,802)aday
      read(ay,802)ayear
      doamdecays=.true.
      if (debug) write(*,*) ' Am month=',am,' month=',amon
      if (debug) write(*,*) ' Am day=',ad,' day=',aday
      if (debug) write(*,*) ' Am year=',ay,' year=',ayear
      if (debug) write(*,*) ' Now doing decay correction'

      call c2jdat(smon,sday,syear,sdays)
      call c2jdat(pmon,pday,pyear,pdays)
      pdelt=sdays-pdays*365.25*(syear-pyear)
      call c2jdat(amon,aday,ayear,adays)
      adelts=sdays-ndays*365.25*(syear-ayear)

```

```

554 continue

```

```

c -----
c ---[ Plutonium Decay Correction ]---
c -----
      old241=wpct(4)
      wttotal = 0.
      if (dopudecay) then
        do 1233 j=1,5
          call decay(wpct,pdelt,thlf(j))
          wttotal = wttotal + wpct(j)
1233  if (wttotal.le.0.)wttotal=1.
        do 1244 j=1,5
          wpct(j)= 100. * wpct(j)/wttotal
1244  p242 = wpct(5)
        endif
c -----
c ---[ Americium Decay Correction ]---
c -----
      if (doamdecay) then
        xpon = exp(-.693147 * adelts / thlf(6))
        wpct(6) = wpct(6)*xpon - 1.034223*old241*(exp(-.693147*
1  1440. * adelts/thlf(4)) - xpon)
      end if

      if (dopudecay) then
        nodecays=.false.
      else
        nodecays=.true.
      endif
555 continue

```

```

c -----
c ---[ End of decay correction ]---
c -----
      if (p242.ne.0) then
        p242fg=.true.
      else
        p242fg=.false.
      endif
      if (nref(1).eq.0 .or. nref(2).eq.0) then
        write(ndev,*) ' Zero value for reference peak. Check input.'
        return
      end if
      GAIN=(STD(2)-STD(1))/(NREF(2)-NREF(1))
      ZERO=STD(2)-NREF(2)*GAIN
      PE242 = PE242 * P242/100.

c
c      APPROXIMATE SHAPE CONSTANTS FOR FWHM AND TAIL PARAMETERS
c
      SHAPC(1)=.2
      SHAPC(2)=.0017
c THESE ARE APPROXIMATE VALUES
      ATAIL=.058
c THIS IS AN APPROXIMATE VALUE FOR GAIN
      BTAIL=.2*GAIN
c
c      INITIALIZE COUNTERS AND POINTERS
c
c      NUMBER OF GROUPS
      NGRP=NRGM
c PEAK COUNTER
      NPIS=0
c STRING POINTER
      NSTRPT=1
c RATIO POINTER
      NRATPT=1
c INITIALIZE 375-KeV PEAK AREA
      A375=0.
c USED FOR FLAG LATER
      AR375 = .FALSE.

c
c      USING REFERENCE PEAKS ESTABLISH GAIN, ZERO, AND PEAK SHAPE
c      PARAMETERS FOR GIVEN SPECTRUM. TWO PASSES ARE MADE BECAUSE
c      OF THE APPROXIMATE ESTIMATES OF PARAMETERS USED ON THE FIRST
c      PASS.
c
      DO 11 JJ=1,2
        J = 1
        DO 10 K=2,6,4

c
c      THE K COUNTER GOES FROM 2 TO 7 TO GET THE SPECIFICATIONS
c      FROM ENSTCH AND ENECH ARRAYS FOR THE 148- AND 208-KEV REGIONS.
c
        IST=(ENSTCH(K)-ZERO)/GAIN
        eend = enech(k)
        if (k.eq.6) eend = enech(k+1)
        IEND=(Eend-ZERO)/GAIN+1
        ST=FLOAT(IST)
        START=ST-1.
        NMS=NMG1(K)
        NME=NMG2(K)
        NODPTS=IEND-IST+1
        NOATA=NODPTS*NMS*NME
        IST=IST-NMS

```

```

      NE1=NDPIS+NUMS
      IF (NDATA.GT.210)THEN
        WRITE(NDEV,9520)
9520  + FORMAT(' Required array size exceeds dimension.
      + Increase gain.')
      write (*,*) 'Press Enter to continue.'
      read (*,*)jnk
      goto 999
      END IF
      DO 9 LM = 1, NDATA
        Y(LM) = DATA(1ST+LM-1)
      C
      C
      C
      DETERMINE BACKGROUND BEFORE AND AFTER PEAK GROUPING
      CALL AVE(1,NUMS,BGD,AVBG1,Y)
      CALL AVE(NE1+1,NDATA,YSUM,AVBG2,Y)
      C
      C
      C
      REMOVE BACKGROUND FROM PEAK GROUPING
      CALL BKGRO(NUMS+1,NE1,AVBG1,AVBG2,0.,0.,NDPIS,SHY,Y,Y)
      C
      C
      C
      THE FOLLOWING ARE APPROXIMATE VALUES FOR DETERMINATION
      OF PEAK LOCATIONS AND WIDTHS USING THE 148.567 AND 208.
      KEV PEAKS
      GSO=GAIN*GAIN
      SG=SHAPC(1)/GSO + SHAPC(2)*(NREF(1)/GAIN + ZERO/GSO) + 0.462
      C
      C
      C
      SG = THE (FULL-WIDTH-HALF-MAXIMUM IN CHANNELS)**2
      ALFA=-2.7726/SG
      C
      C
      C
      GET PEAK CHANNEL POSITION AND WIDTH PARAMETER ALFA
      TEST ERROR VALUE AERR, IT WILL BE SET TO -1 IF PEAK IS NOT
      FOUND OR DOES NOT APPEAR TO BE GAUSSIAN. OTHERWISE AERR
      RETURNS THE ERROR ON ALFA.
      IST = 1 + SE(K)/GAIN
      if (k.eq.6)ist = 4.5/gain
      CALL MAXVAL(IST,NDPIS,JPK,YMAX,Y)
      CALL GF12(JPK,ALFA,ATAIL,BTAIL,PKPS,PHT,AERR,Y)
      if(AERR.LE.-1.)then
        write(ndev,*)'BAD PEAK POSITION FOR REFERENCE PEAK'
        if (printer.eq.'S')then
          write(*,*)'Press Enter to continue'
          read(*,*)jnk
          goto 999
        endif
      endif
      if (alfa.ge.0.) then
        write(ndev,*)'BAD PEAK WIDTH PARAMETER; CHANNEL OVERFLOW'
        if (printer.eq.'S')then
          write(*,*)'Press Enter to Continue'
          read(*,*)jnk
          goto 999
        endif
      endif
      refch(j)=st+pkps
      if (j.eq.1) goto 110
      C(J) = ALFA
      C
      C
      C
      GET PEAK TAILING PARAMETERS ATAIL AND BTAIL. AN APPROXIMATE
      VALUE MUST BE SENT FOR ATAIL IN THE CALL.

```

```

      CALL CTAIL(1,NE1,PKPS,PHT,ALFA,ATAIL,BTAIL,Y)
      J = J + 1
      C
      C
      C
      CONTINUE
      C
      C
      C
      CALCULATE NEW GAIN AND ZERO VALUES FROM DETERMINED PEAK
      POSITIONS
      GAIN=(STD(1)-STD(2))/(REFCH(1)-REFCH(2))
      ZERO=STD(2)-GAIN*REFCH(2)
      SZERO=ZERO
      GSO=GAIN*GAIN
      if (j.eq.1) goto 11
      C
      C
      C
      CHECK ATAIL AND BTAIL PARAMETERS; IF OUT OF RANGE, USE DEFAULT
      VALUES
      IF(BTAIL/GAIN.LT.1.5)BTAIL = 1.5*GAIN
      IF(BTAIL/GAIN.GT.3.5)BTAIL = 3.5*GAIN
      IF(ATAIL.GT.0.5*BTAIL)ATAIL=BTAIL/10.
      C
      C
      C
      REEVALUATE PEAK WIDTH PARAMETER SHAPC(1)
      C(2)=(-2.7726/C(2)-0.462)*GSO
      C(1)=(-2.7726/C(1)-0.462)*GSO
      SHAPC(2)=(C(2) - C(1))/59.5
      IF (SHAPC(2) .LT. .0019) SHAPC(2)=.0017
      IF (SHAPC(2) .GT. .0019) SHAPC(2)=.0019
      SHAPC(1)=C(1)-STD(1)*SHAPC(2)
      11
      C
      C
      C
      CONTINUE
      C
      C
      C
      OUTPUT RESULTS
      ZERO = ZERO + 2.*GAIN
      NT208 = PHT
      FWHM = GAIN * Dsqrt(-2.7726/ALFA)
      WRITE(NDEV,909)FNAME
      write(ndev,911)spectdate,spectime,data(1),data(2)
      C
      C
      C
      .....
      C ...[ Sum up all the channels ]...
      C
      C
      C
      sumall=0
      do 93 i=3,4100
93  sumall=sumall+data(i)
      write(ndev,912)sumall
      write(ndev,400)sumall
      if (dopudecay)write(ndev,913)pudate
      if (dosmdecay)write(ndev,914)smdate
      call More(iline,Printer)
      write(ndev,9101)nref(1)
      call More(iline,Printer)
      write(ndev,9102)nref(2)
      call More(iline,Printer)
      WRITE(NDEV,910)GAIN,ZERO,FWHM
      CALL More(iline,Printer)
      ZERO = SZERO
      ILINE = 8
      if (rep.ne.'N')then
        WRITE (NDEV,9400)SHAPC(1),SHAPC(2),ATAIL,BTAIL

```

```

C      end if
      iline=30
      CALL More(iline,Printer)
      END IF

C      CHECK GAIN,ZERO AND FWHM. IF INVALID, WARN USER.
C      IF BAD GAIN OR ZERO VALUES ARE DETECTED, STOP THE ANALYSIS.
C      IF THE 208-KEV RESOLUTION IS GREATER THAN 1.15 KEV,
C      WARN THE USER BUT CONTINUE THE ANALYSIS.

      IF(GAIN*NUMCHS+ZERO.LE.210.)THEN
        write(ndev,*)'BAD GAIN; CHECK PEAK POSITIONS'
        if (printer.eq.'S')then
          write(ndev,*)'Press Enter to Continue'
          read(*,901)jnk
        endif
        goto 999
      END IF

C      IF(ZERO.GE.121.5)THEN
        write(ndev,*)'BAD ZERO; CHECK PEAK POSITIONS'
        if (printer.eq.'S')then
          write(*,*)'Press Enter to Continue'
          read(*,901)jnk
        endif
        goto 999
      END IF

C      IF(FWHM.GE.1.15)THEN
        write(ndev,*)'POOR RESOLUTION'
        if (printer.eq.'S')then
          write(*,*)'Press Enter to continue'
          read(*,901)jnk
        endif
      END IF

C      LOOK FOR PILEUP ON HIGH-ENERGY SIDE OF 208-KEV PEAK

      PSUM = 0.
      WOFF = PKPS + 1.2/GAIN
      DO 92 J = 1,3
        PSUM = PSUM + Y(WOFF+J)/HT208
      92 IF (PSUM.GT.0.05)THEN
        WRITE (NDEV,9500)
        FORMATT(/,' Significant pileup effects found in spectrum.',/
        * ' Results may be suspect. Check count rate and pole zero.',/)
        END IF

C      LOOK FOR LARGE SCATTERING PEAK BELOW 60-KEV PEAK

      IF (ZERO .LT. 35.)THEN
        IST = (40. - ZERO)/GAIN
        NDPIS = 11./GAIN
        DO 94 I = 1,ndpis
          Y(I) = DATA(IST+I-1)
          CALL MAXVAL(I,NDPIS,JK,YMSC,Y)
          YMSC = YMSC + Y(I)
          IF (YMSC .GT. 3.*SORT(Y(1)))THEN
            IST = (55. - ZERO)/GAIN
            NDPIS = 7./GAIN
            DO 95 I = 1,NDPIS
              Y(I) = DATA(IST+I-1)
              CALL MAXVAL(I,NDPIS,JK,YMAX,Y)
              YMAX = YMAX + Y(I)
              IF (YMAX.GT.3.*SORT(Y(1)).AND.YMSC .GT. YMAX/2.)THEN
                WRITE(NDEV,9600)

```

```

9600 format(' Large scattering peaks observed in spectrum.',/
* ' Analysis will be performed, but results may be suspect.',/
* ' Check measurement setup. Remove objects causing scattering.
* ')
      END IF
      END IF
      END IF

C      BEGIN FITTING EACH PEAK GROUPING

C      if (rep.ne.'N')then
        WRITE(NDEV,921)
      endif
      CALL More(iline,Printer)

      K = 0
      DO 20 JK=1,NGRP
c---[ Put up working on group ... (line) ]---
        if (.not.((rep.eq.'Y').and.(printer.eq.'S'))) then
          call locate(24,0,ier)
          write(*,9500)jks
          endif

          K = K + 1

C      IF U-235 PRESENT, 203-208 KEV PEAKS WERE ANALYZED AS ONE GROUP.
C      SKIP ANALYZING 208-KEV PEAK SEPARATELY
C      IF (K.EQ. 7 .AND. UFG) THEN
        K = K+1
      END IF

C      IF THE SPECTRUM INCLUDES THE 300-KEV REGIONS ADJUST GAIN
C      FOR ANY NONLINEARITY BEFORE CONTINUING THE ANALYSIS

      IF(K.EQ.9)THEN
        STD(2)=375.04
        IST=(STD(2)-ZERO)/GAIN-30
        IF(IST<60.GT.NUMCHS)GOTO 50
        DO 12 LM = 1,60
          Y(LM) = DATA(IST+LM-1)
        12 ST=IST
          SG=SHAPC(1)/GSO + SHAPC(2)*((IST/GAIN + ZERO/GSO) + 0.462
          ALFA = -2.7726/SG
          IST=15
          CALL MAXVAL(IST,60,JK,YMAX,Y)
          CALL GFIT2(JPK,ALFA,ATAIL,BTAIL,PKPS,DHT,AERR,Y)
          IF(AERR.EQ.-1.)then
            write(ndev,*)'BAD 375.0 keV PEAK'
            if(printer.eq.'S')then
              write(*,*)'Press Enter to Continue'
              read(*,901)jnk
            endif
            goto 999
          end if
          GAIN=(STD(2)-208.)/(PKPS+ST -REFCH(2))
          REFCH(2)=PKPS+ST
          ZERO=STD(2)-GAIN*REFCH(2)
        END IF

C      IF U-235 PRESENT, ANALYZE 203-208 KEV REGION AS ONE GROUP
C      IF (K.EQ.6 .AND. A105 .GT. 0.) THEN
        ENSTCH(K)=201.5
        ENRCH(K)=219.7
        NGC2(K) =10

```

```

      UFLG = .TRUE.
      END IF
      IST=(ENSTCH(K)-ZERO)/GAIN
      IEND=(ENECH(K)-ZERO)/GAIN+1
      SG=SHAPC(1)/GSO + SHAPC(2)*(IST/GAIN+ZERO/GSO) + 0.462
      ALFA=-2.7726/SG
      ST=FLOAT(IST)
      START=ST-1.
      NUMS=NBG1(K)
      IF(NUMS.GT.1ST)NUMS=IST-2
      NME=NBG2(K)
      NDPTS=IEND-IST+1
      NDATA=NDPTS+NUMS+NME
      IST=IST-NUMS-2
      NE1=NDPTS+NUMS

      CHECK WHETHER WINDOW IS CONTAINED WITHIN THE SPECTRUM

      IF (NDATA .GT. 210)THEN
        WRITE(NDEV,9520)
        RETURN
      END IF
      IF(IST+NDPTS.GT.NUMCHS)GOTO 50
      DO 13 LM = 1,NDATA
        Y(LM) = DATA(IST+LM-1)
      13 CONTINUE

      STRIP 208-keV TAIL FROM 203-keV peak REGION, IF ANALYZING
      EACH PEAK SEPARATELY, I.E. U-235 NOT PRESENT

      IF (K.EQ. 6 .AND. .NOT. UFLG) THEN
        PKLC = (208. - ZERO)/GAIN - IST
        EXP1 = ATAIL
        EXP2 = BTAIL
        xtemp=0.
        CALL STRIP1(1,NDATA,PKLC,HT208,ALFA,EXP1,EXP2,xtemp,Y)
      END IF

      DETERMINE AREA OF 152-keV PEAK BY SIMPLE INTEGRATION OR
      DETERMINE AREA OF 203-KEV AND 208-keV PEAK BY SIMPLE INTEGRATION,
      IF U-235 IS NOT PRESENT

      IF((K.EQ. 3) .OR.
        (K.EQ. 6) .AND. (.NOT. UFLG) .OR.
        (K.EQ. 7) .AND. (.NOT. UFLG)) THEN
        NSTRP1 = NSTRP1 + 2
        NPKS = NPKS + 1
        PKE = eng(npks)
        NBND5 = Dsqrt( -7.6/ALFA)
        IF (K.EQ.7) NBND5=NBND5*2
        CALL MAXVAL (NUMS,NE1,JKP,YMAX,Y)
        ME = JPK * NBND5
        IF(K.EQ.7)ME=ME*6
        MS = ME * NBG1(K)+1
        ms = jpk * nbnds
        me = ms + nbg2(k)
        IF (MS .LE. 0)MS = 1
        IF (me.le.0)me=ms+1
        MS = JPK * NBND5
        ME = MS + NBG2(K)
        IF (me.gt. ndata) me=ndata
        IF (ms.gt.ndata)ms=me-1

      CALL NCNTHS(MS,ME,MS,ME,ND,S1,AREA(NPKS),ERR(NPKS),AVBG,Y,Y)
      IF(arean(npks).eq.0)arean(npks)=err(npks)

```

```

      ERR(NPKS) = 100. * ERR(NPKS)/AREAN(NPKS)
      IF(rep.ne.'N') then
        WRITE(NDEV,922)X
        CALL MORE(iline,Printer)
        WRITE (NDEV,920)
        CALL MORE(iline,Printer)
      endif

      IF (ERR(NPKS) .GT. 100.) then
        ERR(NPKS) = 100.
      end if

      IF(rep.ne.'N') then
        Call WTemp(snd,pke,area,err,npks,ndev,iline,printer)
      endif
      GOTO 123
    END IF

    IF (K.EQ.6 .AND. UFLG)THEN
      C INTEGRATE ENTIRE 203-208 PEAK REGION AND PROPORTION TOTAL NET
      C AREA ACCORDING TO PEAK HEIGHTS
      C
      CALL NCNTHS(1,NUMS,ME1,NDATA,ND,BGD,TAREA,ER,AVBG,Y,Y)

      ist = 5./gain
      CALL MAXVAL(IST,ND,JKP,YMAX,Y)
      CALL GFIT2(JPK,ALFA,ATAIL,BTAIL,PK208,HT(2),AERR,Y)
      NS = JPK - 1.95/GAIN - 2
      ME = NS + .65/GAIN
      PHT = Y(JPK-2.7/GAIN)
      PKLC = PK208 - 2.685/GAIN
      xtemp = 0.
      CALL STRIP(NS,ME,PKLC,PHT,ALFA,ATAIL,BTAIL,xtemp,Y)
      BG = 0.
      DO 14 I = NS,ME
        BG = BG + Y(I)
      14 CONTINUE
      BG = BG/Float(ME-NS+1)

      C DETERMINE INDIVIDUAL PEAK HEIGHTS
      C
      SUMHT = HT(2)
      DO 15 J = 1,4
        IF (J.EQ.2)GOTO 15
        ALPK = PK208 - (208. + ENG(J+26))/GAIN + .5
        ALFA = ALFA
        CALL GFIT2(JPK,ALFA,ATAIL,BTAIL,PKPS,HT(J),AERR,Y)
        HT(J) = HT(J) - BG*(ENG(J+26)-203.4)/4.4
        IF (HT(J) .LT. 0.) HT(J) = 0.
        SUMHT = SUMHT + HT(J)
      15 CONTINUE

      C DETERMINE PEAK AREAS BY PROPORTIONING TOTAL AREA ACCORDING TO
      C PEAK HEIGHTS
      C
      ER = ER + .005 * TAREA
      IF (rep.ne.'N')then
        WRITE(NDEV,922)X
        CALL MORE(iline,Printer)
        WRITE(NDEV,920)
        CALL MORE(iline,Printer)
      end if
      DO 16 J = 1,4
        NPKS = NPKS + 1
        FRATIO = HT(J) /SUMHT
        AREA(NPKS)=TAREA*FRATIO
        PKE = ENG(NPKS)

```

```

16 IF (AREA(NPKS).LE.0.)THEN
    ERR(NPKS)=100.
    GOTO 16
END IF
ERR(NPKS) = 100. * ER * SQRT(FRATIO)/AREA(NPKS)
if (rep.ne.'N')then
    CALL WRTEHP(snd,pke,area,err,nps,ndev,iline,printer)
end if
18 CONTINUE
NSTRPT = NSTRPT + 4
GO TO 20
END IF

DETERMINE BACKGROUND BEFORE AND AFTER PEAK GROUPING

CALL AVE(1,NUMS,BGD,AVBG1,Y)
CALL AVE(NE1+1,NDATA,YSUM,AVBG2,Y)
IF (K.WE.1) GO TO 103

GET LOW-ENERGY BACKGROUND FOR 125-129 GROUP BETWEEN PEAKS
NUMSE = 4.4/GAIN + NUMS
YMIN = 1.E10
DO 102 I = NUMSE-NUMS,NUMSE
    IF (Y(I) .LT. YMIN)YMIN=Y(I)
    AVBG1 = YMIN
102 CONTINUE

ist=(122.1 - zero)/gain
start = float(ist)-1.
NUMSE = 2.3/GAIN + NUMS + 2
numse= numse-2

103 nest for minimum background in front of 160-Kev peak grouping
if (k.ne.4) goto 21
nest = 1.8/gain + NUMS
call ave(nest-4,nest,bgd,avbg,y)
avbgx = avbg1
if (avbg .lt. avbg1)avbgx=avbg
avbg1 = avbgx

Remove background from peak grouping
21 CALL BKGRD(NUMS+1,NE1,AVBG1,AVBG2,0.,0.,NDPTS,SMV,Y,Y)
AVEBG=0.5*(AVBG1+AVBG2)

GET POSITION OF PEAK OF INTEREST
IST = 1 + SE(K)/GAIN
CALL MAXVAL(IST,NDPTS,JP,K,YMAX,Y)
IF (K.EQ.5 .AND. ndpts-jpk.lt. .9/gain)then
    PKPS=NDPTS/2
    GO TO 23
endif
if (k.eq.5 .and. ymax .lt. 10.*dsqrt(y(1)+avbg1))then
    pkps=ndpts/2
    goto 23
endif
22 CALL GFITZ(JPK,ALFA,ATAIL,RTAIL,PKPS,PKT,AERR,Y)
IF (K.EQ.4 .AND. PKT.LT. 500.)THEN
    PKPS = (164.597 - ZERO)/GAIN * START + 1.
END IF

```

```

IF (PKT .GT. 500.)THEN
    IF (ALFA .GT. -.5)SG = -2.7726/ALFA
END IF

CHECK THE POSITION OF THE 129.3-KEV PEAK. MAKE SURE
THE CORRECT REFERENCE PEAK POSITIONS WERE USED.

IF(K.WE.1)GOTO 23
REFEN=(PKPS+START)*GAIN + ZERO
IF(ABS(REFEN-129.3).GT.1.)then
    write(ndev,*)'CHECK REFERENCE PEAK POSITION'
    if (printer.eq.'S')then
        write(*,*)'Press Enter to Continue'
        read(*,901)jnk
    endif
    goto 999
end if

SETUP PEAK REFERENCE FLAGS FOR FITTING
23 WP=(char(STR(NSTRPT))-48
    IF (K.EQ.4)npks=npks+1

NSET = 0
IF (K .EQ.6 .AND. UFLG) THEN
    NP = 4
    NSET = 1
END IF

IF (K .EQ. 8 .AND. .NOT. UFLG) NPKS = NPKS + 2
IF (K .EQ. 8 .AND. UFLG)UFLG = .FALSE.

DO 25 J=1,NP
    NXFLG(J)=(char(STR(NSTRPT+J))-48
    PKMT(J)=1.
    IF (UFLG .AND. J.EQ.2) NXFLG(J) = NP
    IF(NXFLG(J).NE.NP)GOTO 24
    REFEN=ENG(J+NPKS)
    NXFLG(J)=NSET
24 IF(NXFLG(J).NE.NSET)THEN
    PKMT(J)=RATIO(WRATPT)
    WRATPT=WRATPT + 1
END IF
IF (UFLG) NXFLG(J) = 0
25 CONTINUE

NSTRPT=NSTRPT+NP+1
IF (UFLG) NSTRPT = NSTRPT - 1

ZERO ARRAYS
DO 26 I=1,6
    SUM(I)=0.
    NO 26 J=1,6
26 ALPHA(I,J)=0.
    DO 27 J=1,6
        AX(I,J)=0
27 CONTINUE

CALCULATE NEW GAIN FOR REGION
SGAIN = GAIN
IF((K.EQ.5).OR.(K.EQ.7).OR.(K.EQ.10))GOTO 28
IF (K .EQ. 6 .AND. UFLG)GO TO 28
SGAIN=(REFEN-STD(2))/(PKPS+START-1.-REFEN(2))
DO 29 J=1,NP

```

```

29 PKPOS(J)=(ENG(J*PKPS)-REFEN)/SGAIN*PKPS
C
C GENERATE PEAK RESPONSE PROFILE FOR PEAK GROUPING AND STORE
C IN THE 'AX' ARRAY.
C
DO 30 I=1,NDPTS
  WT=1.0/(Y(I)+AVEBG)
  XI=1
  L=1
  DO 35 JJ=1,NP
    DLTX=XI-PKPOS(JJ)
    DLXSQ=DLTX*DLTX
    ALFA=-2.7726/(SHAPC(2)*(PKPOS(JJ)-PKPS)/GAIN + SG)
    EXPNT = ALFA*DLXSQ
    CB=0.
    IF(DLTX.LT.0.)THEN
      EXPH=BTAIL*DLTX
      IF(BTAIL*DLTX,GE.-10.)then
        EXPN2=EXP(BTAIL*DLTX)
        EXPN3=EXP(0.4*EXPNT)
        CB=BTAIL*EXPN2*(1.-EXPN3)
      END IF
    ELSE
      IF(NXFLG(JJ).GT.0)THEN
        AX(I,NXFLG(JJ))=AX(I,NXFLG(JJ))*PKHT(JJ)*(EXP(EXPNT)+CB)
      ELSE
        AX(I,L)=AX(I,L)+PKHT(JJ)*(DEXP(EXPNT)+CB)
        L=L+1
      END IF
  35 CONTINUE
  NFREE=L-1
  DO 350 KK=1,NFREE
    SUM(KK)=SUM(KK)+AX(I,KK)*Y(I)*WT
  350 CONTINUE
  DO 350 KL=1,NFREE
    ALPHA(KK,KL)=ALPHA(KK,KL)+AX(I,KK)*AX(I,KL)*WT
  350 CONTINUE
  INVERT MATRIX TO GET PEAK HEIGHTS
  CALL MATINV(ALPHA,6,NFREE,DET)
  CALCULATE PEAK HEIGHTS FROM INVERTED MATRIX ELEMENTS
  DO 360 L=1,NFREE
    PT(L)=0.
    DO 360 J=1,NFREE
      PT(L)=PT(L)+SUM(J)*ALPHA(L,J)
  360 CONTINUE
  CHISQ=0.
  DO 46 I=1,NDPTS
    YCALC=0.
    DO 45 L=1,NFREE
      YCALC=YCALC+PT(L)*AX(I,L)
  45 RES=Y(I)-YCALC
  WT=Y(I)+AVEBG
  IF (WT .LE. 0)WT = 1.
  RM=RES/Dsqrt(WT)
  CHISQ=CHISQ+RM**2
  CHAN=START +I
  C The next line had a D in the first column
  C WRITE (6,940)CHAN,Y(I),RES,RM
  46 CONTINUE
  CHISQ=CHISQ/NDPTS

```

```

C
C SAVE CORRELATION BETWEEN 125- AND 129-KeV PEAKS
C
C IF (K.EQ.1) THEN
C   EC129 = ALPHA(2,3)*3.1416 * (-1./ALFA)
C   MD IF
C
C PRESERVE CORRELATION BETWEEN THE 159.955- AND 160.3-KEV
C PEAKS FOR ERROR ANALYSIS LATER
C IF (K.EQ.4) THEN
C   EC160 = ALPHA(1,2)*3.1416 * (-1./ALFA)
C   EC165 = ALPHA(1,5)*3.1416 * (-1./ALFA)
C   END IF
C
C SAVE CORRELATION BETWEEN 203- AND 208-KeV PEAKS
C WHEN U-235 IS PRESENT
C IF (K.EQ.6 .AND. UFLG) THEN
C   EC208 = ALPHA(1,2) * 3.1416 * (-1./ALFA)
C   END IF
C
C SAVE CORRELATION BETWEEN 368- AND 375-KeV PEAKS
C FOR ERROR ANALYSIS LATER
C IF (K.EQ.10) THEN
C   EC375 = ALPHA(1,4) * 3.1416 * (-1./ALFA)
C   END IF
C
C if (rep.no.'M') then
C   WRITE (NOEV,922)K
C   CALL More(1,1,Printer)
C   WRITE (NOEV,925)CHISQ
C   CALL More(1,1,Printer)
C   WRITE(NOEVI,920)
C   CALL More(1,1,Printer)
C   end if
C
C CALCULATE PEAK ENERGIES, AREAS, AND ASSOCIATED ERRORS
C
C IF(K.EQ.1 .OR. K.EQ.4) CHISQ=1.
C IF (CHISQ.LT.1.)CHISQ=1.
C L=1
C DO 40 J=1,NP
C   NPXS=NPKS+1
C   ERR(NPKS)=0.
C   ALFA = -2.7726/(SHAPC(2)*(PKPOS(J)-PKPS)/GAIN + SG)
C   IF(NXFLG(J))J=1,61,62
C   PKHT(J)=PKHT(J)*PT(L)
C   IF(PKHT(J).LE.0)PKHT(J)=1.
C   if (alpha(I,L).LE.0)alpha(I,L)=0.
C   ERR(NPKS)=Dsqrt(ALPHA(L,L))*3.1416 * (-1./ALFA)
C   L=L+1
C   GOTO 43
C   PKHT(J)=PKHT(J)*PT(NXFLG(J))
C   CHAN = PKPOS(J)+START
C
C   pke = (PKPOS(J)+ START-1..refch(2))*SGAIN + std(2)
C   AREA(NPKS) = PKHT(J)*(1.7724*Dsqrt(-1./ALFA))
C   if (k.eq.5 .and. abs(pke-185.7).gt.0.55)then
C     area(npk)=0.
C     end if
C   if (k.eq.5.and.area(npk).le.4.*err(npk))then
C     area(npk)=0.
C     end if
C   if(area(npk).le.0)then
C     area(npk)=0.
C     err(npk)=100.

```

```

      goto 47
    end if
    err(npks) = 100. * sqrt(chisq)*err(npks) / area(npks)
44    IF(ERR(NPKS).EQ.0) go to 47
    IF (ERR(NPKS) .GT. 100.) then
      ERR(NPKS) = 100.
    end if
47    if (rep.ne.'N') then
      Call WTemp(snd,pke,area,err,npks,ndev,iline,printer)
    endif
50    CONTINUE
C Rodetermine the 332- and 345-keV peak areas
C
123    IF (K.EQ.9) THEN
      AREA(30)=0.
      BG=0.
      NPTS=(334.65 - ENSTCH(9))/GAIN
      DO 48 J=1,NPTS
        BG=BG+AVEBG
48      AREA(30)=AREA(30)+Y(J)
      ER332=100.*SQRT(AREA(30)+2.*BG)/AREA(30)
      AREA(37)=0.
      BG = 0.
      NPTS=(ENECH(9)-343.8)/GAIN
      DO 49 J=1,NPTS
        BG = BG + AVEBG
49      AREA(37)=AREA(37)+Y(J)
      ER345 = 100.*SQRT(AREA(37)+ 2.*BG)/AREA(37)
    ENDIF
C the next line had the 123 label
    IF (K.EQ.10) GO TO 50
    CONTINUE
C
C DIVIDE SELECTED PEAK AREAS BY APPROPRIATE BRANCHING RATIOS
C TO GIVE DISINTEGRATIONS. USE THIS INFORMATION TO DETERMINE
C RELATIVE DETECTION EFFICIENCY.
C
50    NGRP = 4
    NP = 4
C
    if (a208.le.0) then
      write(ndev,*)'208-keV peak area is <= 0. Analysis Aborted'
      if (printer.eq.'S')then
        write(*,*)'Press Enter to Continue'
        read(*,901)jnk
      endif
      goto 999
    end if
    EF208 = A208/5.33E-06
    X(1)=dlog(0.12924)
    Y(1)=(A129/6.26E-05)/EF208
    ERROR(1)=DSQRT(ER129**2+ER208**2)
    X(2)=dlog(0.16458)
    Y(2)=(A165/4.64E-07)/EF208
    ERROR(2)=DSQRT(ER165**2+ER208**2)
    X(3)=dlog(0.203537)
    Y(3)=(A203/5.62E-06)/EF208
    ERROR(3)=DSQRT(ER203**2+ER208**2)
    X(4)=dlog(.208)
    Y(4)=1.
    ERROR(4)=ER208
C
C INCLUDE THE 300-KEV INFORMATION OF BOTH GROUPS WERE ANALYZED

```

```

    IF(A267.EQ.0)GOTO 53
    NGRP=5
    NP=5
    X(5)= dlog(.26754)
    Y(5)= (A267/1.75E-07)/EF208
    ERROR(5) = DSQRT(ER267**2+ER208**2)
    IF (A375 .EQ. 0 OR .A370 .EQ. 0)GOTO 53
    X(6)=dlog(.345014)
    Y(6)=(A345/5.59E-06)/EF208
    ERROR(6)=DSQRT(ER345**2+ER208**2)
    X(7)=dlog(.375042)
    Y(7)=(A375/1.55E-05)/EF208
    ERROR(7)=DSQRT(ER375**2+ER208**2)
C
C # OF PEAKS TO BE USED; A375 NONZERO
    NGRP = 7
C # OF FREE PARAMETERS TO BE USED IN FIT
    NP = 5
C 300-keV regions included
    AN375 = .TRUE.
C
C Check peak area for negative or zero values before fit
C
53    do 54 j=1, ngrp
      if (y(j).le.0.) then
        write(ndev,970) j,an(nj)
        if (printer.eq.'S')then
          write(*,*)'Press Enter to Continue'
          read(*,901)jnk
        endif
        goto 999
      end if
54    continue
C
C CARRY OUT LOG-LOG FIT
C
55    CALL PUFFIT(NP,NGRP,CHISO)
C
C TEST PU-238 152-KEV PEAK AREA. IF ZERO, SET TO A SMALL VALUE
C AND SET ERROR TO 100%.
C
    IF(A152.EQ.0) THEN
      A152 = 10.
      ER152= 100.
    END IF
C
C CALCULATE NEW ERROR ON 160-KEV PEAK MULTIPLET. INCLUDE CORRELATION
C BETWEEN 159.955- AND 160.3-KEV PEAKS.
C
57    ERR(19)=(ERR(17)/100.*AREA(17))**2+
    1 (ERR(19)/100.*AREA(19))**2 + 2. * EC160
    IF (ERR(19) .LE. 0)THEN
      ERR(19) = ERR(19) * 2. * EC160
    END IF
    ERR(19) = SQRT(ERR(19))
    XTP1=.1595
    XTP2=.1604
    IXTIP2=2
    FACT = DPH(XTP2,IXTP3)/DPH(XTP1,IXTP3)
    A160 = AREA(17)*FACT*AREA(19)
    IF (A160 .LE. 0)THEN
      A160 = 10.
      ER160= 1.
    END IF
    ERR(19) = 100. * ERR(19)/A160

```



```

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      CALCULATE ISOTOPIC RATIOS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      ESTABLISH WEIGHT FACTORS WTF
C
      xtemp=.1293
      XTEMP1=.1253
      IXT3=2
      WRATIO = (A125/(DPM(XTEMP, IXT3)*EF208))**2
      ARATIO = (A129*DPM(XTEMP1, IXT3))/(DPM(XTEMP, IXT3)*A125)**2
      WTF(1) = WRATIO * ARATIO * ((ER125**2+ER129**2)/100.**2
      -2. * EC129/(A129*A125))
1
      xtemp=.1527
      XTEMP1=.1486
      IXT3=2
      WRATIO = (A148/(DPM(XTEMP, IXT3)*EF208))**2
      ARATIO = (A152*DPM(XTEMP1, IXT3))/(DPM(XTEMP, IXT3)*A148)**2
      WTF(2) = WRATIO * ARATIO * ((ER148**2+ER152**2)/100.**2)
1
      xtemp=.1604
      XTEMP1=.1646
      IXT3=2
      WRATIO = (A165/(DPM(XTEMP, IXT3)*EF208))**2
      ARATIO = (A160*DPM(XTEMP1, IXT3))/(DPM(XTEMP, IXT3)*A165)**2
      WTF(3) = WRATIO * ARATIO * ((ER160**2+ER165**2)/100.**2
      -2. * EC165/(A160*A165))
1
      xtemp=.2036
      XTEMP1=.208
      IXT3=2
      WRATIO = (A208/(DPM(XTEMP, IXT3)*EF208))**2
      ARATIO = (A203*DPM(XTEMP1, IXT3))/(DPM(XTEMP, IXT3)*A208)**2
      WTF(4) = WRATIO * ARATIO * ((ER203**2+ER208**2)/100.**2
      -2. * EC208/(A203*A208))
1
      ZERO ARRAYS
58 DO 60 J =1,4
      RAT(J) = 0.
      ERR(J) = 0.
      BETAJ(J) = 0.
59 DO 59 JJ =1,4
      ALPHA(J, JJ) = 0.
      DO 60 K =1,7
      Y(K) = 0.
      P(K, J) = 0.
60
      LOAD MATRIX ELEMENTS
      DPM IS DEFINED AS A FUNCTION WHICH CALCULATES A PSEUDO
      DISINTEGRATION NUMBER AT THE ENERGY AND FOR THE ISOTOPE
      SPECIFIED. A RATIO OF THESE VALUES IS EQUIVALENT TO
      THE RATIO OF THE RELATIVE DETECTION EFFICIENCIES AT THE
      ENERGIES INDICATED.
      EQUATION FOR THE 125- AND 129-KeV PEAKS
      xtemp1=.12529
      xtemp2=.12929
      IXT3=2
      FACT=A129 * DPM(xtemp1, IXT3)/(A125*DPM(xtemp2, IXT3))
      P(1,4)= FACT * 3.99E-5
      Y(1) =6.26E-5

```

15

```

C
      EQUATION FOR THE 148- AND 152-KeV PEAKS
C
      P(2,1)= 9.26E-6
      xtemp1=.14857
      xtemp2=.1527
      IXT3=2
      FACT = A152 *DPM(xtemp1, IXT3)/(A148*DPM(xtemp2, IXT3))
      P(2,3)= -FACT*1.90E-6
C
      EQUATION FOR THE 160-KeV PEAK GROUPING AND THE 165-KeV PEAK
C
      P(3,2)= -.4,02E-6
      xtemp1=.1646
      xtemp2=.1604
      IXT3=2
      FACT=A160*DPM(xtemp1, IXT3)/(A165*DPM(xtemp2, IXT3))
      P(3,3)=FACT * 4.58E-7 - 6.48E-8
      P(3,4)=FACT * 4.641E-7 - 6.661E-8
      P(3,4)=FACT * 6.26E-7
      Y(3) = 0.
C
      EQUATION FOR THE 203- AND 208-KeV PEAKS
C
      xtemp1=.208
      xtemp2=.2036
      IXT3=2
      P(4,3)=A203*DPM(xtemp1, IXT3)/(A208*DPM(xtemp2, IXT3))
      P(4,4)=P(4,3) * 7.51E-6 - 2.9E-8
      P(4,3)=P(4,3) * 5.33E-6
      r(4) = 5.6E-6
C
      INCLUDE INFORMATION FROM THE 300-KeV PEAKS IF THEY ARE
      PRESENT IN THE SPECTRUM.
C
      IF(.NOT.AR375 .OR. A370 .EQ. 0)GOTO 262
      xtemp=.333
      XTEMP1=.345
      IXT3=2
      WRATIO = (A332/(DPM(XTEMP, IXT3)*EF208))**2
      ARATIO = (A345*DPM(XTEMP, IXT3))/(DPM(XTEMP, IXT3)*A332)**2
      WTF(5) = WRATIO * ARATIO * ((ER332**2+ER345**2)/100.**2)
      xtemp=.375
      XTEMP1=.370
      WRATIO = (A370/(DPM(XTEMP, IXT3)*EF208))**2
      ARATIO = (A375*DPM(XTEMP1, IXT3))/(DPM(XTEMP, IXT3)*A370)**2
      WTF(6) = WRATIO * ARATIO * ((ER370**2+ER375**2)/100.**2)
      -2. * EC375/(A370*A375))
1
C
      EQUATION FOR THE 330-KeV PEAK GROUPING
C
      xtemp1=.333
      xtemp2=.345
      IXT3=2
      FACT = A345*DPM(xtemp1, IXT3)/(A332*DPM(xtemp2, IXT3))
      P(5,3) = FACT * 2.977E-7
      P(5,4)= FACT * 1.49E-6
      Y(5) = 5.59E-6 - FACT* 5.054E-6
C
      EQUATION FOR THE 370-KeV PEAK GROUPING
C
      xtemp1=.370

```

```

xtemp2=.375
ixtp3=2
FACT = A375 * DPM(xtemp1,ixtp3)/(A370 * DPM(xtemp2,ixtp3))
P(6,3) = FACT * 1.074E-8
P(6,4) = FACT * 2.17E-6
Y(6) = 1.57E-5

      SETUP MATRIX FOR ISOTOPIC RATIOS
      IF(NGRP.EQ.4)NGRP=5
      DO 65 I=1,ngrp-1
        DO 65 J=1,4
          BETA(J) = BETA(J) + P(I,J) * Y(I) / WTF(I)
        DO 65 K=1,4
          ALPHA(J,K) = ALPHA(J,K) + P(I,J)*P(I,K) / WTF(I)
      END DO
      INVERT MATRIX TO GET ISOTOPIC RATIOS

      CALL MATINV(ALPHA,6,4,DET)

      CALCULATE ISOTOPIC RATIOS AND ERRORS FROM THE INVERTED
      MATRIX ELEMENTS

      DO 70 J=1,4
        ERR(J) = Dsqrt(ALPHA(J,J))
      DO 70 K=1,4
        RAT(J) = RAT(J) + BETA(K) * ALPHA(J,K)

      CORRECT RATIOS FOR HALF-LIVES AND MASS DIFFERENCES

      Z4849=RAT(1) * .003628
      EZ4849= ERR(1)* .003628
      Z4049=RAT(2) * .27252
      EZ4049= ERR(2)* .27252
      Z4149 = RAT(3) * .0006009
      EZ4149= ERR(3) * .0006009
      Z5149 = RAT(4) * .018025
      EZ5149= ERR(4) * .018025
      EC4840 = ALPHA(1,2) * .003628 * .27252
      EC4841 = ALPHA(1,3) * .003628 * .0006009
      EC4041 = ALPHA(2,3) * .27252 * .0006009
      IF (A160.EQ.10.) THEN
        Z4049=0.00010
        EZ4049=0.00010
        EC4840=0.
        EC4041=0.
      write(ndev,9700)
9700 format(' Pu-240 160-keV area = 0. Results are suspect. ')
      IF (PRINTER .EQ. 'S') THEN
        WRITE(NDEV,*) 'Press Enter to Continue'
        READ (*,901)JNK
      ENDIF
      END IF

      EC4851 = ALPHA(1,4) * .003628 * .018025
      EC4051 = ALPHA(2,4) * .27252 * .018025
      EC4151 = ALPHA(3,4) * .0006009 * .018025
      Z5141 = Z5149 * 1. / Z4149
      EZ5141 = EZ5149 * 1. / Z4149

      EZ5141=Z5141*Dsqrt((EZ5141/Z5141)**2+(EZ4149/Z4149)**2
      + 2. * EC4151/(Z4149*Z5149))

```

```

xtemp=.18572
ixt1=1
DPM239 = A203/5.62E-6 * DPM(xtemp,ixt1)
Z2549= A105 * 7.038E08/(.561*DPM239*Z4119.)
EZ2549 = Dsqrt(ER105**2 + ER203**2)*Z2549/100.

      COMPARE DETERMINED RATIOS WITH OPERATORS DECLARED RATIOS
      USING T**2 STATISTICAL TEST

      if (.not.noddecay) CALL TTEST(WPCT,RAT,NGRP,12)

      CALCULATE TIME SINCE SEPARATION OF AM-241 BASED ON THE
      AM-241/PU-241 RATIO

      This was a LOG, not DLOG
      PUAGE=Dlog(0.96694*Z5141+1.)/1.2793E-04
      EPUAGE=(.96694*EZ5141)/((1.+0.96694*Z5141)*1.2793E-04)

      Output last page of results
      WRITE(NDEV,200)FNAME
      iline=iline + 1
      CALL More(iline,Printer)

      WRITE(NDEV,201)Z4849,EZ4849,Z4049,EZ4049,
      1 Z4149,EZ4149,Z5141,EZ5141,Z2549,EZ2549
      IF (.REP.NE. 'M') THEN
        iline=iline+12
        CALL More(iline,Printer)
      END IF
      WRITE(NDEV,202)PUAGE,EPUAGE,PUAGE/365.25,EPUAGE/365.25
      iline=iline+1
      CALL More(iline,Printer)

      DETERMINE THE PU-242 ABUNDANCE AND CALCULATE THE WEIGHT PERCENT
      ABUNDANCE FOR EACH PU ISOTOPE. CORRECT PU-241 ABUNDANCE FOR
      DECAY SINCE SEPARATION. NOTE: THIS CORRECTION IS NOT COMPLETE
      BECAUSE IT IGNORES COOLING TIME.

      DETERMINE THE PU-242 ABUNDANCE AND CALCULATE THE WEIGHT PERCENT
      ABUNDANCE FOR EACH PU ISOTOPE. CORRECT PU-241 ABUNDANCE FOR
      DECAY SINCE SEPARATION IN ADDITION TO AN ESTIMATED COOLING TIME
      OF 4 YEARS. THE ERROR CALCULATION ASSUMES A 100% ERROR ON
      THE 4 YEAR ESTIMATE.

      P242X = 0.
      IF(P242FG)GOTO 80
      X4149=Z4149*EXP((PUAGE/365.25+COLTIM)*0.048292)
      P242=52.*Z4049*X4149
      P242X = P242
      P242= P242 * SORT(((EZ4049/Z4049)**2 + (EZ4149/Z4149)**2
      + 2. * EC4041/(Z4049*Z4149) + 2.0E-8 + 1.740E-8*EPUAGE**2)
      pe242x = pe242

      80
      YS=1.+Z4849+Z4049+Z4149
      IF (P242X.NE.0)PE242X = 0.000000
      P239=(100.-P242)/YS
      PE239 = P239/YS * SORT(PE242X**2/P239**2 + EZ4849**2 +
      1 (1.+P242X/(P239*Z4049))**2+EZ4049**2 +
      2 (1.+P242X/(P239*Z4149))**2+EZ4149**2 +
      3 2.*(1.+P242X/(P239*Z4049))*EC4840 +
      4 2.*(1.+P242X/(P239*Z4149))*EC4841 +
      5 2.*(1.+P242X/(P239*Z4149))*(1.+P242X/(P239*Z4049))*EC4041)

```

17

File: 1A.FOR

```

if(pz24fg)then
    calc='1'
else
    calc='Calculated'
end if
write(ndev,301)Z42,pZ42,peZ42,calc
ELSE
WRITE(NDEV,211)
WRITE(NDEV,213)
WRITE(NDEV,214)
WRITE(NDEV,305)P38,P39,PE38,WPC(1),ORATIO(1)
WRITE(NDEV,305)P39,P39,PE39,WPC(2),ORATIO(2)
WRITE(NDEV,305)P40,P40,PE40,WPC(3),ORATIO(3)
WRITE(NDEV,305)P41,P41,PE41,WPC(4),ORATIO(4)
IF (pz24fg)THEN
WRITE(NDEV,305)Z42,PZ42,PEZ42,WPC(5),ORATIO(5)
ELSE
CALC='Calculated'
END IF
END IF

CALL More(Iline,Printer)
iline=iline+25

if (wpc(2).eq.0)then
WRITE(NDEV,205)aZ41,aE241,U235,uE235
CALL More(Iline,Printer)
else
write(ndev,206)aZ41,aE241,WPC(6),oratio(6),u235,uE235
call More(Iline,printer)
end if

*****
CALCULATE SPECIFIC POWER AND PU-Z40 EFFECTIVE

CALL SPPOW(PCT,RAT,POW,POME,PZ4FG,PZ42,PEZ42,EZ4O,EZ4OE)

WRITE (NDEV,207)EZ4O,EZ4OE
CALL More(Iline,Printer)
WRITE (NDEV,209)POW,POME
CALL More(Iline,Printer)
CALL More(Iline,Printer)
CALL More(Iline,Printer)
IF (I2 .EQ. -1) THEN
WRITE (NDEV,210)
write(ndev,"*)
GO TO 990
END IF
WRITE (NDEV,212)I2
CALL More(Iline,Printer)

if(nodecay)then
    write(ndev,*")
    write(ndev,*")
    * "No analysis date(s) entered, decay correction ",
    * "and t**2 test not performed."
end if

990 IF (PRINTER.eq.'S') THEN
write(' ')Press Enter to Continue"
read(*,901)jnk
write(*,'*)Ok"
end if

c-...{ Pack Common Variables for writing to AMRESULT.DBF }-----
cfname=sfame
cp238=p238
```

```

cp239=pe239
cp240=pe240
cp241=pe241
cp242=pe242
cpe238=pe238
cpe239=pe239
cpe240=pe240
cpe241=pe241
cpe242=pe242
cpow=pow
cpowe=powe
ce240=e240
ce240e=e240e
csample=sample
cal(1)=alpha(1,1)
cal(2)=alpha(1,2)
cal(3)=alpha(1,3)
cal(4)=alpha(1,4)
cal(5)=alpha(2,2)
cal(6)=alpha(2,3)
cal(7)=alpha(2,4)
cal(8)=alpha(3,3)
cal(9)=alpha(3,4)
cal(10)=alpha(4,4)
cz4849=z4849
cez4849=cz4849
cz4049=z4049
cez4049=cez4049
cz4149=z4149
cez4149=cez4149
cz5141=z5141
cez5141=cez5141
cz2549=z2549
cez2549=cez2549
csumall=sumall
ca241=a241
cae241=ae241
cu235=u235
cue235=ue235
cfwhm=fwhm

```

```

C ...[ If sample was supplied, write results do database ]...

```

```

    if (samflag) then
        call anresult
    endif

```

```

C ...[ Do a page eject if printer selected ]...

```

```

999    if (printer.eq.'P')write(ndev,800)

        close(ndev)
        call cis
        return
    END

```

\$large

Subroutine Cicero

CICERO
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : CICERO
Revision : 1.6
Author : Russell Fleming
Date : 12/15/87
Language : Microsoft Fortran Ver. 4.01

Abstract : This program acquires spectra data from a Silena
Multichannel analyzer and writes the data to a
Binary file.

Arguments

Routines
Called : None

Libraries
Used : Filesub, dbsubs, Grafsub

Revision History (Include date, revision I.D., and details)

COMMON and variable declarations

```
character*1 byte,x,ack,exflag,key
character*8 data(0:9000), temp*6
character*8 id,lt
character*15 dbname,jnk
character filename*45
character*11 names(127,2)
integer*2 Baud,parity,stopbit,Databits,comport,ier,rstat,
```

```
+ status,max,count,sbyte,unit,lens(127,2),nrec,outer
+ ,ltinc,ldinc
integer*4 Wdata(0 : 9000),finfo(4)
real*4 ddata(127)
character*127 cdata(127)
logical*2 gotsome
```

variable initialization declarations

Initialize Comm Parameters
9600,E,7,2 Comm1:

```
baud=7
parity=2
stopbit=2
Databits=7
comport=:0
ack=char(6)
gotsome=.FALSE.
```

format declarations

```
34 format(a45)
17 format(' Estimated transfer time: 40 Seconds.')
18 format(' Now receiving data ... ',9)
98 format(' Transfer completed with',i6,' channels sent')
65 format(i6)
66 format(i8)
151 format('////////////////////')
```

```
174 -- time
```

C Get file name

```
open(4,file='cicero.prm',iostat=istat,status='OLD')
read(4,34)filen
close(4)
```

C Write file name to temporary file to DBASE

```
open(4,file='FILENAME.CIC',iostat=istat,status='UNKNOWN')
if(istat.ne.0)then
write(*,*)
write(*,*)'Error writing data file name -- Press Enter'
read(*,34)jnk
stop
end if
write(4,34)filen
close(4)
```

C Start data transfer

```
call atbaud(baud,parity,stopbit,Databits,comport,rstat,ier)
if(ier.ne.1)then
write(*,*)'Error in Setup Parameters -- Press Enter'
read(*,34)jnk
end if
call comon
```

```

      x = char(110)
      Call SndBt(x,comport,status)
      if (status.lt.0) goto 500
      if (status.eq.-1) goto 510

C --- delay loop
10  do 10 i=1,500
      continue

      x = char(112)
      Call SndBt(x,comport,status)
      if (status.lt.0) goto 500
      if (status.eq.-1) goto 510

      write(*,*)
      write(*,17)
      write(*,18)

C  Get Channel Data
222 i=1
111 i=i+1
      do 40 j=1,8
         count=0
25  call bstat(sbyte)
         if (sbyte.eq.0) then
            count=count+1
            if (count.gt.10000) goto 80
            goto 25
         endif
         count=0
         if (sbyte.eq.-1) goto 550
         call fchchr(byte,status)
         write(*,*)'byte ',byte,' j ',j
         if (status.ne.1) goto 25
         if (byte.eq.'i') goto 80
         if (byte.eq.ack) goto 25
         data(i)(j):=byte
         gotsome=.TRUE.
40  continue
      goto 111

80  continue
      call offint

      if (.not. gotsome) then
         write(*,151)
         write(*,*)
         write(*,*)' Error in receiving data from the analyzer.'
         write(*,*)' -----'
         write(*,*)
         write(*,*)' Make sure that the analyzer is connected properly!'
         write(*,*)' and is in the correct mode for transfer.'
         write(*,*)
         write(*,*)' If the analyzer is connected properly, and is in'
         write(*,*)' the correct mode, then the analyzer cannot detect'
         write(*,*)' the commands coming from the computer. You will'
         write(*,*)' need to manually set the analyzer to send data out'
         write(*,*)' to the computer.'
         write(*,*)
         write(*,*)' Enter an 'M' to manually set the analyzer or'
         write(*,*)' a 'R' to retry data transfer or any'
         write(*,*)' other key to abort and return to main menu'
         write(*,*)
         call inkey(key)
         if ((key.eq.'R').or.(key.eq.'i')) goto 222
         if ((key.eq.'M').or.(key.eq.'m')) then

```

```

         write(*,*)
         write(*,*)' Manual data transfer from analyzer!'
         write(*,*)' -----'
         write(*,*)
         write(*,*)' To manually transfer data from the analyzer,'
         write(*,*)' connect the analyzer properly, then Press Enter'
         write(*,*)' on the computer to start the transfer. Then Press'
         write(*,*)' the Data Out button on the analyzer to send the '
         write(*,*)' data from the analyzer to the computer.'
         write(*,*)
         write(*,*)' Press Enter on the computer to start the transfer,'
         write(*,*)' then set the analyzer to send data out.'
         call inkey(key)
         goto 222
      end if
      write(*,151)
      goto 543
      end if

      write(*,98)
      if (i.gt.4100) write(*,*)' Only 4100 channels recorded.'

      count=i

      if (count.lt.3) goto 543
      write(*,*)
      write(*,*)' Now writing data file'

      open(4,file=filen,status='UNKNOWN',form='BINARY')

C  Get Spectrum ID and Livetime
      outer=1
      inc=0
C  Get ID
      ID=1
1300 ID(1:8)=data(1)(1:8)

C  Get Live time
      LT=1
1400 LT(1:8)=data(2)(1:8)

C  Convert ID and Livetime to data to write to binary file
      read(id(3:8),65)ndata(0)
      write(4)ndata(0)
      read(lt(3:8),65)ndata(1)
      write(4)ndata(1)

      write(*,*)' Spectrum ID: ',id
      write(*,*)' Livetime : ',lt

C  Write data to file
      if (count.gt.4098)count=4098

C
      do 100 i=3,count
         do 200 j=3,8
            temp(j-2:j-2)=data(i)(j:j)
200  continue
C      write(*,*)temp
         read(temp,65)ndata(i)
         write(4)ndata(i)
100  continue

C  ... [ Dropped channel analysis ] ...
C

```

```

do 400 i=2,count
  if ((i.gt.3).and.(ndata(i-1).ge.20).and.(ndata(i+1).ge.20)
  *   .and.(ndata(i).eq.0))then
    write(*,*)
    write(*,*)'WARNING',char(7)
    write(*,*)'Zero count channel found at channel number',i
    write(*,*)'This spectra file may not be usable. ',
  *   'Retry data transfer.'
    write(*,*)
    write(*,*)'Press Enter to Continue'
    read(*,34)jnk
    goto 400
  endif
400 continue
  goto 543

500 call offint
  write(*,151)
  write(*,*)' Error in issuing commands to analyzer'
  write(*,*)' .....'
  write(*,*)' The computer is unable to send commands to the'
  write(*,*)' analyzer. Check the cabling connecting the'
  write(*,*)' computer and analyzer.'
  write(*,*)
  write(*,*)' Press an 'R' to retry the transfer or any other'
  write(*,*)' key to abort transfer.'
  call inkey(key)
  if ((key.eq.'R').or.(key.eq.'r'))goto 123
  goto 543

510 call offint
  write(*,151)
  write(*,*)' Port address Error'
  write(*,*)' .....'
  write(*,*)' There is an error in addressing Com port 1.'
  write(*,*)' Verify that there is a serial port in the computer.'
  write(*,*)
  goto 543

550 write(*,*) char(13),'BUFFER OVERFLOW -- Transfer aborted',char(7)
  write(*,*)'Remove any memory-resident software and try again.'
  call offint
  write(*,*)

543 write(*,*)' Press Enter to continue.'
  read(*,34)jnk
  return
99  End
....>

```

```

1 large

```

```

      subroutine anresult

```

```

      ANRESULT
      =====

```

```

      Developed for : Safeguards Technology Program
                    : Nuclear Chemistry Division
                    : Lawrence Livermore National Laboratory
                    : Livermore, CA. 94550

```

```

      This work supported by LLNL POW 9107605

```

```

      Title       : ANRESULT
      Revision    : 1.1
      Author      : Russell Fleming
      Date        : 1/11/88
      Language    : Microsoft Fortran Ver. 4.01

```

```

      Abstract : Save IAEA analysis variables to anresult.dbf dBase file

```

```

      Arguments :

```

```

      Routines
      Called   : None

      Libraries
      Used     : None

```

```

      Revision History ( include date, revision I.D., and details )

```

```

      COMMON and variable declarations

```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```

      $INCLUDE: 'top.cmn'
      common /top/ drivepath
      character drivepath*45

```

```

      $INCLUDE: 'iaesdb.cmn'
      common /dbsave/ cp238,
                     + cpe238,
                     + cp239,
                     + cpe239,

```

```

      c + cp240,
      c + cpe240,
      c + cpe241,
      c + cpe242,
      c + cpe242,
      c + cpow,
      c + cpowe,
      c + ce240,
      c + ce240e,
      c + cfname,
      c + csample,
      c + cal,
      c + cz4849,
      c + ce24849,
      c + cz4049,
      c + ce24049,
      c + cz4149,
      c + ce24149,
      c + cz5141,
      c + ce25141,
      c + cz2549,
      c + ce2549,
      c + csamal,
      c + ca241,
      c + cae241,
      c + cu235,
      c + cue235,
      c + cfwhm

```

```

      character cfname*30,csample*10
      dimension cal(10)

```

```

      variable initialization declarations

```

```

      dimension fnames(127,2),flens(127,2),finfo(4),ndata(127),
      + cdata(127)

```

```

      character fnames*11,cdata*127,flen*45,dp*45,fn*30,nwfn*30
      character ch*1

```

```

      real*4 ndata

```

```

      integer*2 flens,lunit,nrcc,il, strien
      integer*4 finfo

```

```

      logical debug

```

```

      debug=.false.

```

```

      if (debug) write(*,*) 'Now in anresult'
      if (debug) write(*,*) 'Drivepath=',drivepath,' '

```

```

      dp=drivepath
      if (debug) write(*,*) 'Dp=',Dp,' '

```

```

      il=strlen(dp)
      if (debug) write(*,*) 'Length of "',dp,'" = ',il
      if (il.le.1) then
         write(*,*) 'Error in default drive and path.'

```



```

      if (debug) write(*,*)'Length of "',dp,'" = ',ii
      if (i.le.1) then
        write(*,*)
        write(*,*)'Error in default drive and path.'
      +   write(*,*) 'Check system defaults from the main menu.'
      endif
      filen=dp(1:ii)//'anresult.dbf'

      if (debug) then
        write(*,*)'In anresult, csample="'csample,'"
        pause
      endif

      iunit=3
      do 20 i=1,127
        fnames(i,1)= '
      20   fnames(i,2)= '

c ...[ Just file name is saved to database, strip off drive and path ]...

      fn=cfname
      llen=gstrlen(fn)
      i=llen
      70   if (i.gt.0) then
            ch = fn(i:i)
            if ( (ch.ne.'\') .and. (ch.ne.':') ) then
              i=i-1
            else
              goto 80
            endif
            goto 70
          endif
      80   if (i.gt.0) then
            newfn=fn(i+1:)
          else
            newfn=fn
          endif

      call dbopen(filn,iunit,fnames,flen,finfo)

      cdata(1) = newfn
      if (debug) write(*,*)'newfn='newfn,

      cdata(2) = csample
      if (debug) write(*,*)'csample="'csample,'"
      ndata(1) = sngl(cp238)
      if (debug) write(*,*)'cp238='cp238
      ndata(2) = sngl(cp239)
      if (debug) write(*,*)'cp239='cp239
      ndata(3) = sngl(cp240)
      if (debug) write(*,*)'cp240='cp240
      ndata(4) = sngl(cp241)
      if (debug) write(*,*)'cp241='cp241
      ndata(5) = sngl(cp242)
      if (debug) write(*,*)'cp242='cp242

      ndata(6) = sngl(cpe238)
      if (debug) write(*,*)'cpe238='cpe238
      ndata(7) = sngl(cpe239)
      if (debug) write(*,*)'cpe239='cpe239
      ndata(8) = sngl(cpe240)
      if (debug) write(*,*)'cpe240='cpe240

```

```

      ndata(9) = sngl(cpe241)
      if (debug) write(*,*)'cpe241='cpe241
      ndata(10) = sngl(cpe242)
      if (debug) write(*,*)'cpe242='cpe242

      ndata(11) = sngl(cpow)
      if (debug) write(*,*)'cpow='cpow
      ndata(12) = sngl(cpowe)
      if (debug) write(*,*)'cpowe='cpowe

      ndata(13) = sngl(ce240)
      if (debug) write(*,*)'ce240='ce240
      ndata(14) = sngl(ce240e)
      if (debug) write(*,*)'ce240e='ce240e

      do 10 i=1,10
      10   ndata(14+i) = sngl(cal(i))
            if (debug) write(*,*)'cal('14+i,')='ndata(14+i)

      ndata(25) = sngl(cz4849)
      if (debug) write(*,*)'ndata(25)='cz4849

      ndata(26) = sngl(cez4849)
      if (debug) write(*,*)'ndata(26)='cez4849

      ndata(27) = sngl(cz4049)
      if (debug) write(*,*)'ndata(27)='cz4049

      ndata(28) = sngl(cez4049)
      if (debug) write(*,*)'ndata(28)='cez4049

      ndata(29) = sngl(cz4149)
      if (debug) write(*,*)'ndata(29)='cz4149

      ndata(30) = sngl(cez4149)
      if (debug) write(*,*)'ndata(30)='cez4149

      ndata(31) = sngl(cz5141)
      if (debug) write(*,*)'ndata(31)='cz5141

      ndata(32) = sngl(cez5141)
      if (debug) write(*,*)'ndata(32)='cez5141

      ndata(33) = sngl(cz2549)
      if (debug) write(*,*)'ndata(33)='cz2549

      ndata(34) = sngl(cez2549)
      if (debug) write(*,*)'ndata(34)='cez2549

      ndata(35) = sngl(csumal1)
      if (debug) write(*,*)'ndata(35)='csumal1

      ndata(36) = sngl(ca241)
      if (debug) write(*,*)'ndata(36)='ca241

      ndata(37) = sngl(cae241)
      if (debug) write(*,*)'ndata(37)='cae241

      ndata(38) = sngl(cu235)
      if (debug) write(*,*)'ndata(38)='cu235

      ndata(39) = sngl(cue235)
      if (debug) write(*,*)'ndata(39)='cue235

      ndata(40) = sngl(cfwhm)
      if (debug) write(*,*)'ndata(40)='cfwhm

```

```
      nrec = finfo(4)+1
      call dbwrite(filen,iunit,fnames,flens,finfo,cdata,ndata,nrec)

      if (debug) then
        write(*,*)'Leaving anresult'
        pause
      endif

99      return
      end
```

\$large

SUBROUTINE AVE (NS,ME,SUM,AV,Y)

```

      AVE
      =====

```

Developed by : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

This work supported by LLNL POW 9107605

```

Title       : AVE
Revision    : 2.2
Author      : R. Gunnink
             : Revised for the IBM PC by Russell Fleming
Date        : 10/15/86
Language    : Microsoft Fortran   Ver. 4.01

```

Abstract :
 This routine determines the average count of a group of data values from NS to ME inclusive. It also returns the sum.

```

Arguments :
INPUT
NS      Starting position for group in Y array
ME      Ending position for group in Y array
Y       Data array

OUTPUT
AV      Average count of data points in group set by NS and ME
SUM     Sum of data points in group

```

Routines
 Called : None

Libraries
 Used : None

Revision History (Include date, revision I.D., and details) *

Original Version: 19-Jan-1978, Ver. 2.1

COMMON and variable declarations *

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

DIMENSION Y(210)

```

C *****
C      variable initialization declarations
C *****

```

```

C      SUM = 0.0
C *****

```

```

C      SM = ME - NS + 1
C      DO 2 I = NS,ME
C          SUM = SUM + Y(I)
C      AV = SUM / SM
C      RETURN
C      END

```

\$large

SUBROUTINE BKGD (NS,NE,AV1,AV2,S2,ND,Y1,Y2)

```

      BKGD
      =====

```

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL PGN 9107605

```

      Title      : BKGD
      Revision   : 2.1
      Author    : Ray Gunnink
      Date      : 1/28/88
      Language  : Microsoft Fortran Ver 4.01

```

Abstract :

To calculate the background continuum under a peak region, to subtract it from the channel counts, and return the net channel and total counts.

Like NCTS, this subroutine calculates the background distribution for a specified peak region, performs a channel-by-channel subtraction, and returns the net area and net count array. However, the input arguments provide the before and after average background levels to be used. The routine uses one pass for a positive sloping background, and two for a negative slope. The first pass determines a straight line connecting the before and after background levels. The second pass additionally computes a smoothed-step background which is subtracted from the gross channel counts to obtain the net count array and the integrated net count.

This routine is similar to the NCTS and BKGRD routines. The BKGRD is more rigorous and also takes into account the entrance and exit slopes of the background.

Arguments :

INPUT
NS = Beginning channel of peak grouping
NE = Ending channel of peak grouping
AV1= Average background level in front of the peak grouping.
AV2= Average background level aft of the peak grouping.
Y1 = Array containing gross counts of the peak grouping.

RETURN

ND = Number of channels in peak grouping
S2 = Net counts in the peak region.
Y2 = Array of net counts generated by the routine.

```

C OTHER VARIABLES
C DBG = Change in background level.
C CHANS = Number of channels in the peak grouping.
C SLOPE = Average change in background per channel.
C AVSIG = A factor related to the statistical variance of the data
C         in the region. It is used to provide smoothing to
C         statistically poor data.
C SUMY = Variable used to accumulate the net count using
C         the straight line for background determination.
C SUM = The final value of SUMY following the 1st pass.
C YN = Net count of a channel (temporary argument).
C YNT= Positive value only of YN (temporary argument).
C SLBG = Straight line background value at a given channel.

```

```

      Routines
      Called : None

```

```

      Libraries
      Used : None

```

Revision History (Include date, revision I.D., and details) *

Version 2: 12-Oct-84

COMMON and variable declarations *

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      dimension Y1(210), Y2(210)
      INTEGER*2 NS,NE,ND

```

variable initialization declarations *

```

C Change in background level.
      DBG = AV2 - AV1
      ND = 0
      S2 = 0.0
C Total channels in group.
      CHANS = NE - NS + 1
C Average change in background per channel.
      SLOPE = DBG / CHANS
      IF (AV2.LE.0)AV2=1.
C Smoothing factor
      AVSIG = 5./CHANS*SQRT(AV2*CHANS)
      NN = 2
C One pass for positive sloping bkgd.
      IF (SLOPE .GT. 0.0) NN = 1
      DO 30 J = 1,NN
          SUMY = 0.0
C Initial straight line bkgd value set at AV1.
          SLBG = AV1
      DO 20 I = NS,NE
C New straight line bkgd value.
          SLBG = SLBG + SLOPE
C Calc. net count.

```

```
      DO 20 I = NS,NE
C   New straight line bkgrd value.
      SLBG = SLBG + SLOPE
C   Calc. net count.
      YN = Y1(I) - SLBG
      YNT = YN
C   Only integrate positive values.
      IF (YN .LT. 0.0) YNT = 0.0
C   Running sum of net counts.
      SUMY = SUMY + YNT + AVSIG
C   Straight line only on + slope
      IF (SLOPE .GT. 0.0) GOTO 10
C   Straight line only on 1st pass
      IF (J .EQ. 1) GOTO 20
C   Calc. smoothed-step bkgrd value.
      YN = Y1(I) - AV1 - DBG*SUMY/SUM
10      ND = ND + 1
C   Store net channel count.
      Y2(ND) = YN
C   Integrate net channel counts.
      S2 = S2 + YN
20      CONTINUE
C   Store total net counts using straight line bkgrd.
      SUM = SUMY
C   Get out if SUM is zero or less
      IF (SUM .LE. 0.)RETURN
30      CONTINUE
      RETURN
      END
```

\$large

SUBROUTINE BKGRD (NS,NE,BGLO,BGHI,SLPLO,SLPHI,NPTS,SNV,Y,YNET)

BKGRD

=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL POW 9107605

Title : BKGRD
Revision : 1.0
Author : Ray Gunnink
Revised for the IBM PC by Russell Fleming
Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :

This subroutine calculates the background distribution for a given peak region, subtracts it, and returns the net count array. Like NCTS and BKGD, this routine generates an accurate step-shaped background. However, in these routines, BKGRD takes into consideration the sloping backgrounds surrounding the region of interest. If the entrance and exit slopes differ, the routine assumes that the change from one slope to the other is monotonic.

Arguments :

INPUT

NS = starting channel of the region
NE = ending channel of the region.
BGLO = background count in channel next to low E side of peak region.
BGHI = background level next to high energy side of region.
SLPLO = background slope on low side (in counts/channel)
SLPHI = background slope on high side (in counts/channel)
Y(I) = array containing original spectral data.

RETURN

NPTS = number of channels in YNET array.
SNV = sum of net counts returned to calling program
YNET(I) = array into which net counts are placed (may be same as Y).

OTHER VARIABLES

DBG = Change in background level.
SIG & AVSIG = Factors related to the statistical variance of the data in the region. They are used to provide a measure of smoothing to statistically poor data.
SLPBG = Contribution to the bkgrd due to the slope of the bkgrd.
SLBG = Straight line background value at a given channel.
YN = Net count of a channel.
YNT = Positive value only of YN.
SUMY1 = Integrated net area as obtained using a straight line bkgrd.

SUMY2 = Integrated net area as obtained using a step-function bkgrd.
YSUM1 = Variable used to accumulate straight-line net area.
YSUM2 = Variable used to accumulate step-function net area.
YS1,YS2,YS3,YS4 = Accumulator variables used to account for histogram nature of the data. ie use halves of 2 adjacent channels.

Routines
Called : None

Libraries
Used : None

Revision History (Include date,revision I.D., and details) *

Version: 1b
Author: Ray Gunnink
Date: 12-Dec-83
Last revision: 20-Sep-84 (R.G.)

COMMON and variable declarations *

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
REAL*8 BGLO,BGHI,SLPLO,SLPHI,SNV,Y(210),YNET(210)
INTEGER*2 NS,NE,NPTS

variable initialization declarations *

Number of channels in grouping.
NPTS = NE-NS+1
XPTS = NPTS
Level change in bkgrd
DBG = (BGHI-BGLO) -0.5*(SLPLO+SLPHI)*(XPTS+1.)
Average bkgrd slope (in counts/channel)
SLOPE = DBG / (XPTS + 1.)
Starting (low-side) bkgrd level.
SLBG = BGLO
SUMY1 = 0.0
YSUM2 = 0.0
IF (BGLO.LE.0.) BGLO = 1.
Variable related to statistical level of bkgrd
SIG = 5.*SQRT(BGLO)
AVSIG = SIG/XPTS
L = 0
SLPBG = 0.0

Get 1st estimate of net area of peak(s).
DO 10 I = NS,NE
Fractional distance through grouping.
FRAC = L / XPTS
Amt due to sloping bkgrd.
SLPBG = SLPBG+SLPLO*(1.-FRAC)+SLPHI*FRAC
Straight line background level.
SLBG = SLBG + SLOPE
L = L + 1
Net count in channel using straight line

```

      SLPBG = SLPBG+SLPLO*(1.-FRAC)+SLPHI*FRAC
C  Straight line background level.
      SLBG = SLBG + SLOPE
      L = L + 1
C  Net count in channel using straight line
      YN = Y(I) - SLBG - SLPBG
C  Only accumulate positive values.
      IF (YN .LT. 0.) YN = 0.
C  Approximate area using straight line.
10      SUMY1 = SUMY1 + YN
C  Add value related to statistical variance.
      SUMY1 = SUMY1 + SIG
      NN = 2
C  Only 1 pass if slope is positive.
      IF (DBG .GE. -10.0) NN = 1

C  1st uses straight line. 2nd also uses step.
      DO 30 J = 1,NN
      L = 0
C  Start at initial bkgrd level.
      SLBG = BGLO
      YSUM1 = 0.0
      SLPBG = 0.0
C  Data is histogram. Use half of 2 adjacent channels.
      YS1 = 0.5 * BGLO
      YS3 = YS1
      SMY = 0.0

      DO 20 I = NS,NE
C  Straight line background.
      SLBG = SLBG + SLOPE
C  Fractional distance through grouping
      FRAC = I / XPTS
C  Background slope.
      SLPBG = SLPBG + SLPLO *(1.-FRAC)+SLPHI *FRAC
C  Net count using straight line.
      YN = Y(I) - SLBG - SLPBG
      YNT = YN
C  Only integrate positive values.
      IF (YN .LT. 0.) YNT = 0.
      L = L + 1
C  Use straight line only for + step.
      IF (DBG .GE. -10.) GOTO 14

C  Partial area up to channel "I".
      YSUM1 = YSUM1 + YNT + AVSIG
C  Other half.
      YS2 = 0.5 * (BGLO + DBG + YSUM1/SUMY1)
C  Net count using straight line bgnd.
      YN = Y(I) - YS1 - YS2 - SLPBG
C  Skip following on first iteration.
      IF (J .EQ. 1) GOTO 15

C  Partial area to channel "I" using step bgnd.
      YSUM2 = YSUM2 + YN + AVSIG
C  Other half.
      YS4 = 0.5 * (BGLO + DBG + YSUM2/SUMY2)
C  Net count resulting from refined step.
      YN = Y(I) - YS3 - YS4 - SLPBG
      YS3 = YS4
C  Transfer to YNET array on last iteration.
14      YNET(L) = YN
C  Net counts (or area).
15      SMY = SMY + YN

      YS1 = YS2

```

```

20      CONTINUE
C  New value of SUMY2 for 2nd pass.
      SUMY2 = SMY + SIG
30      CONTINUE

      RETURN
      END

```

\$large

SUBROUTINE BWF (X)

```

C *****
C
C                               BWF
C                               =====
C *****

```

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL POF 9107605

```

C *****

```

Title : BWF
Revision : 1.0
Author : Wayne D. Ruhter
Revised for the IBM PC by Russell Fleming
COMPUchem, Inc.
3099B Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

```

C *****

```

Abstract :

```

C *****

```

Arguments : X

```

C *****

```

Routines
Called : None
Libraries
Used : None

```

C *****

```

Revision History (Include date, revision I.D., and details) *

```

C *****

```

COMMON and variable declarations *

```

C *****

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION CA(5),CB(6)

```

C *****

```

variable initialization declarations *

```

C *****

```

```

C *****

```

```

CA(1) = .729281
CA(2) = .0344557
CA(3) = .127639
CA(4) = -.0325239
CA(5) = .00642257
CB(1) = .933301
CB(2) = -4.45603
CB(3) = 1.16191
CB(4) = -.141124
CB(5) = -.0352464
CB(6) = .00744387

```

```

C *****

```

```

C *****

```

```

2      IF (X - 2.2) 2,10,10
      XX = X * X
      AX = 1.
      SUM = 0.0
      DO 4 I = 1,5
        AX = AX * XX
        SUM = SUM + CA(I) * AX
4      X = SUM * EXP (-XX)
      GO TO 20
10     XLOG = LOG (X)
      SUM = CB(1)
      DO 12 I = 2,6
        SUM = SUM + CB(I) * XLOG ** (I-1)
12     X = EXP (SUM)
20     RETURN
      END

```

```

---->

```


\$large

SUBROUTINE C2JDAT(MONTH,NDAY,NYR,DDAY)

```

C*****
C                C2JDAT
C                *****
C*****

```

```

C
C Developed by : Safeguards Technology Program
C               Nuclear Chemistry Division
C               Lawrence Livermore National Laboratory
C               Livermore, CA. 94550
C

```

```

C This work supported by LLNL PO# 9107605
C*****

```

```

C Title       : C2Jdat
C Revision    : 1.0
C Author      : Wayne D. Ruhter
C Date        : 11/9/87
C Language    : Microsoft Fortran Ver. 4.01
C*****

```

```

C Abstract    : Converts calendar month and day to Julian date.
C*****

```

```

C Arguments   : MONTH,NDAY,NYR,DDAY
C*****

```

```

C Routines
C Called      : None
C

```

```

C Libraries
C Used        : None
C*****

```

```

C Revision History ( Include date,revision I.D., and details )
C*****

```

```

C COMMON and variable declarations
C*****

```

```

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C

```

```

C MONTH = Calendar month
C NDAY  = Calendar day
C NYR   = Year
C DDAY  = Julian day of year
C*****

```

```

C variable initialization declarations
C*****

```

```

C DIMENSION NDAYS(12)
C DATA NDAYS/31,28,31,30,31,30,31,31,30,31,30,31/
C

```

```

C DDAY = NDAY
C NMONTH = MONTH - 1

```

```

IF(NMONTH.EQ.0)GOTO 20
DO 10 I = 1,NMONTH
DDAY = DDAY + NDAYS(I)
IF(MONTH.LT.3)RETURN
IF(MOD(NYR,4).EQ.0)DDAY = DDAY + 1.

```

10

20

```

RETURN
END

```

\$large

subroutine DCHECK

```

      DCHECK
      =====

```

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL PDM 9107605

Title : Dcheck
Revision : 1.0
Author : Wayne D. Rutherford
Modified for the IBM PC and MS-DOS by Russell Fleming
COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

Date : 19-NOV-86

Update : 11/10/87 VERSION 1.3.3

Language : Microsoft Fortran Ver. 4.01

Abstract :

This program performs peak-shape analysis on peaks specified and provides user with full-width-half-maximum values at 122- and 208-keV. It also examines the high-energy side of the high-energy peak for pileup. If it is significant, it warns the user. It also checks the gain to determine if it is appropriate for plutonium isotopic measurements.

Arguments :

Routines
Called : None

Libraries
Used : None

Revision History (Include date, revision I.D., and details)

COMMON and variable declarations

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```

      common /dchk/ nref,std,livtim,crate,zero,gain,wfwhm,sfwhm,
     + shape,parea,spara,puflg,fname

```

```

      DIMENSION NREF(2),STD(20),REFCH(2),C(2),Y(200),SHAPC(2),
     + DATA(0:4100),parea(2),spara(2),wfwhm(2),sfwhm(2)

```

```

      logical*2 puflg
      CHARACTER*30 FNAME
      integer*2 numchs,dd(B),temp

```

```

      integer*4 ddata(0:4100),livtim

```

```

C *****
C variable initialization declarations
C *****
C

```

```

C *****
C format declarations
C *****
C

```

```

906 FORMAT(' Cannot open spectrum named: ',A)
909 format(' Livetime: ',i6,' seconds',/,
     + ' Count Rate: ',f9.1,' counts/second',/)
905 + FORMAT(' RESULTS FROM ',/, ' ANALYSIS OF: ',A)
907 FORMAT(' GAIN = ',F7.4,/, ' ZERO = ',F8.3,/, ' FWHM(208 KEV) = ',
     + F5.3,/, ' FWHM(122 KEV) = ',F5.3)
910 FORMAT(/, ' Gain MUST BE LESS than 0.15 keV/channel')
920 FORMAT(/,
     + ' CHECK GAIN: Upper energy range MUST BE AT LEAST 210 keV')
930 FORMAT(/, ' CHECK ZERO: Zero CANNOT BE GREATER than 121.5 keV')
940 FORMAT(/, ' POOR RESOLUTION -- CHECK SYSTEM AND COUNT RATE')
950 FORMAT(/, ' HIGH-ENERGY TAILING IS SIGNIFICANT',/,
     + ' CHECK SYSTEM POLE-ZERO SETTING AND INPUT COUNT RATE.')
```

C OUTPUT DEVICE LOGICAL UNIT

```
DATA NDEV/6/
```

C INPUT DEVICE LOGICAL UNIT

```
DATA INDEV /2/
```

```

C GET POSITIONS AND ENERGIES OF PEAKS TO ESTABLISH GAIN, ZERO, AND
C PEAK SHAPE PARAMETERS

```

```

C GET APPROXIMATE CHANNEL POSITIONS FOR REFERENCE PEAKS
C IF ENTRY IS NEGATIVE, EXIT FROM PROGRAM

```

```

C CALCULATE APPROXIMATE GAIN AND ZERO FOR SYSTEM

```

```
call dchdat(nref,std,fname)
```

```

GAIN=(STD(2)-STD(1))/(NREF(2)-NREF(1))
ZERO=STD(2)-NREF(2)*GAIN

```

```

C OPEN(INDEV,FILE=FNAME,STATUS='OLD',FORM='BINARY',
     + IOSTAT=TEMP)

```

```
if(temp.ne.0)goto 6
```

```
GO TO 7
```

```
write(*,906)FNAME
```

```
goto 999
```

```

C READ DATA FILE INTO DATA ARRAY

```

```

C
7 CALL Rbin(NUMCHS,CDATA,INDEV)
do 1234 i=1,numchs
1234 data(i)=dble(cdata(i))
C
C GET LIVETIME FROM SPECTRUM. EXPECTED IN FIRST CHANNEL.
C
C LIVTIM = idint(DATA(1))
C IF (LIVTIM .EQ. 0)GOTO 110
C
C SUM THE TOTAL SPECTRUM TO DETERMINE THROUGHPUT RATE.
C
C SUM = 0.
C DO 10 J = 1,NUMCHS
10 SUM = SUM + DATA(J)
C RATE = SUM/dble(LIVTIM)
C
C APPROXIMATE SHAPE CONSTANTS FOR FWHM AND TAIL PARAMETERS
C
110 pufig=.false.
SHAPC(1)=.2
SHAPC(2)=.0017
C THESE ARE APPROXIMATE VALUES
ATAIL=.0.058
C THIS IS AN APPROXIMATE VALUE FOR GAIN
BTAIL=.2.0*GAIN
C
C USING REFERENCE PEAKS ESTABLISH GAIN, ZERO, AND PEAK SHAPE
C PARAMETERS FOR GIVEN SPECTRUM. TWO PASSES ARE MADE BECAUSE
C OF THE APPROXIMATE ESTIMATES OF PARAMETERS USED ON THE FIRST
C PASS.
C
C DO 30 JJ=1,2
C J = 1
C DO 20 K = 1,2
C
C DETERMINE STARTING CHANNEL FOR PEAK REGION
C
C IST= NREF(J) - 40
C ST=FLOAT(IST)
C START=ST-1.
C
C SET NUMBER OF CHANNELS TO BE USED FOR BACKGROUND LEVEL MEASUREMENT.
C
C NUMS=2
C NUME=2
C
C READ IN DATA FOR PEAK REGION
C DO 9 LM = 1,80
9 Y(LM) = DATA(IST+LM-1)
C
C DETERMINE MORE PRECISE REGION FOR PEAK
C
C GSO=GAIN*GAIN
C SG=SHAPC(1)/GSO + SHAPC(2)*(NREF(J)/GAIN + ZERO/GSO) + 0.462
C
C SG = THE (FULL-WIDTH-HALF-MAXIMUM IN CHANNELS)**2
C
C ALFA=-2.7726/SG
C
C GET PEAK CHANNEL POSITION AND WIDTH PARAMETER ALFA
C TEST ERROR VALUE AERR, IT WILL BE SET TO -1 IF PEAK IS NOT
C FOUND OR DOES NOT APPEAR TO BE GAUSSIAN. OTHERWISE AERR
C RETURNS THE ERROR ON ALFA.

```

```

ix1=1
ix55=55
CALL MAXVAL(ix1,ix55,JP,K,YMAX,Y)
C
CALL GFIT2(JPK,ALFA,ATAIL,BTAIL,PKPS,PHT,AERR,Y)
C
C if((-.9./ALFA).lt.0.) then
C write(*,*) 'Error - Bad Shape Parameter'
C pause 'Please Press Enter to Continue.'
C GOTO 999
C
C endif
C NBND5 = SORT(-.9./ALFA) + 1.
C NS1 = JPK - NBND5 - NUMS - 4
C NE1 = JPK + NBND5
C NS = NS1
C
C SEARCH FOR FRONT-SIDE MINIMUM
C
C DO 11 I = 1,5
C IF (Y(NS+I)) .GT. Y(NS1)) GO TO 11
C NS1 = NS + 1
11 CONTINUE
C IF (Y(NS1-1)) .LT. Y(NS1+1)) NS1 = NS1 - 1
C NS2 = NS1 + 1
C NS = NE1
C
C SEARCH FOR AFT-SIDE MINIMUM
C
C DO 12 I = 1,5
C IF (Y(NS+I)) .GT. Y(NE1)) GO TO 12
C NE1 = NS + 1
12 CONTINUE
C
C Add on to obtain region for pileup check
C
C NE1 = NE1 + 2
C
C IF (Y(NE1-1)) .LT. Y(NE1+1)) NE1 = NE1 - 1
C NE2 = NE1 + 1
C
C REMOVE BACKGROUND FROM PEAK REGION AND DETERMINE NET PEAK AREA
C
C CALL NCNTS(NS1,NS2,NE1,NE2,NO,BO,PAREA(J),SPAREA(J),avbg,y,Y)
C
C NCNTS = NE1 - NS2
C
C ix1=1
C CALL MAXVAL(ix1,NDPTS,JP,K,YMAX,Y)
C CALL GFIT2(JPK,ALFA,ATAIL,BTAIL,PKPS,PHT,AERR,Y)
C IF(AERR.lt.0.) then
C write(*,*) 'BAD PEAK POSITION'
C pause 'Please Press Enter to Continue.'
C GOTO 999
C
C endif
C WFWHM(J) = ALFA
C SFWHM(J) = SORT(AERR)
C REFCN(J)=ST*PKPS + NS1
C J = J + 1
20 CONTINUE
C
C CALCULATE NEW GAIN AND ZERO VALUES FROM DETERMINED PEAK
C POSITIONS
C
C GAIN=(STO(1)-STD(2))/(REFCH(1)-REFCH(2))
C ZERO=STD(2)-GAIN*REFCH(2)
C GSO=GAIN*GAIN
C

```

```

C      REEVALUATE PEAK WIDTH PARAMETER SHAPC(1)
C
C      C(2)=(-2.7726/MFWHM(2)-0.462)*GSQ
C      C(1)=(-2.7726/MFWHM(1)-0.462)*GSQ
C      SHAPC(2) = (C(2) - C(1))/(STD(2) - STD(1))
C      SHAPC(1)=C(1)-STD(1)*SHAPC(2)
30    CONTINUE
C
C      CHECK HIGH-ENERGY SIDE OF HIGH ENERGY PEAK FOR PILEUP
C
C      NOFF = PKPS + 1.2/GAIN
C      SUM = 0.
C      DO 40 J = 1,3
40    SUM = SUM + Y(NOFF+J)/PHT
C
C      OUTPUT RESULTS TO THE PRINTER
C
C      ZERO = ZERO + 2.*GAIN
C      P122 = (122.05-ZERO)/GAIN
C      SG = SHAPC(1)/GSQ + SHAPC(2)*(P122/GAIN+ZERO/GSQ)+0.462
C      if (sg.lt.0) then
C        write(*,*)' Error - Bad Shape Parameter'
C        pause 'Please Press Enter to Continue.'
C        GOTO 999
C      endif
C      FW122 = SORT(SG)*GAIN
C      P208 = (208.-ZERO)/GAIN
C      SG = SHAPC(1)/GSQ + SHAPC(2)*(P208/GAIN+ZERO/GSQ)+0.462
C      if (sg.lt.0) then
C        write(*,*)' Error - Bad Shape Parameter'
C        pause 'Please Press Enter to Continue.'
C        GOTO 999
C      endif
C      FW208 = SORT(SG)*GAIN
C      WRITE(*,905)FNAME
C      WRITE(*,909)LIVTIM,CRATE
C      WRITE(*,907)GAIN,ZERO,FW208,FW122
C
C      CHECK GAIN,ZERO AND FWHM. IF INVALID, WARN USER.
C      IF BAD GAIN OR ZERO VALUES ARE DETECTED, STOP THE ANALYSIS.
C      IF THE 208-KEY RESOLUTION IS GREATER THAN 1.15 KEV,
C      WARN THE USER BUT CONTINUE THE ANALYSIS.
C
C      IF (GAIN .GT. .15)write(*,910)
C
C      IF(GAIN*NUMCHS+ZERO.LE.210.)THEN
C        write(*,920)
C      END IF
C
C      IF(ZERO.GE.121.5)THEN
C        write(*,930)
C      END IF
C
C      IF(FW208.GE.1.15)THEN
C        write(*,940)
C      END IF
C
C      IF (SUM .GT. .002)THEN
C        puflag=true.
C        write(*,950)
C      END IF
C
C      Save parameters to dBase file
C      call dsave

```

34

```

999    pause 'Please Press Enter to Continue.'
      return
      end
.....

```

\$large

SUBROUTINE CTAIL(NS,NE,PKPOS,PKHT,ALFA,A,B,Y)

CTAIL
=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL POF 9107605

Title : CTAIL
Revision : 1.1
Author : Wayne D. Ruhter
Revised for the IBM PC by Russell Fleming
Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :

This subroutine strips the gaussian component from a peak. Any residual remaining at low energies is treated as short-term tailing. The positive data are fit to a straight line by least-squares to give a short-term tail amplitude and slope.
NOTE: The peak must have good statistics for this subroutine to give good results.

Arguments :

NS = STARTING POSITION OF PEAK IN THE "Y" ARRAY
NE = ENDING POSITION OF PEAK IN THE "Y" ARRAY
PKPOS = PEAK CENTROID IN THE "Y" ARRAY
PKHT = PEAK HEIGHT
ALFA = PEAK WIDTH PARAMETER
A = SHORT-TERM TAIL AMPLITUDE DETERMINED BY THIS ROUTINE
B = SHORT-TERM TAIL SLOPE DETERMINED BY THIS ROUTINE
ENTER SUBROUTINE WITH AN INITIAL ESTIMATE FOR THIS PARAMETER
Y = ARRAY OF PEAK DATA

Routines
Called : None
Libraries
Used : None

Revision History (Include date, revision I.D., and details)

Version 1.0 1-Nov-80

```

C      COMMON and variable declarations
C      *****
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      DIMENSION Y(210)
C      *****
C      variable initialization declarations
C      *****
C      RES =0.
C      A =0.
C      SUMX =0.
C      SUMX2=0.
C      SUMY =0.
C      SUMXY=0.
C      SUMW =0.
C      *****
C      CALCULATE POINT AT WHICH THE TAIL SHOULD START TO ROLL-OFF
C      BEHEATH THE PEAK BASED ON INITIAL ESTIMATE FOR THE TAIL SLOPE.
C      XM = -SQRT(-2.7726/ALFA)*1
C      STRIP OUT THE GAUSSIAN
C      K=1
C      DO 10 I=NS,NE
C      X1 = 1
C      DX = X1 - PKPOS
C      IF(DX.GT.XM)GOTO 11
C      DXSQ = DX * DX
C      ADX = ALFA * DXSQ
C      GAUSSIAN COMPONENT
C      FXK = DEX(ADX)*PKHT
C      RESIDUAL IS TAIL
C      TY = Y(I) - FXK
C      IF(FXK.LT.1.0)GOTO 10
C      IF (RES .NE. 0) GO TO 8
C      IF (Y(I).LE. 0) GO TO 10
C      RES = Y(I) + 3.* SQRT(Y(I))
C      IF(TY.LT.1.0)GOTO 10
C      IS RESIDUAL STILL SIGNIFICANT
C      IF(Y(I).GT.RES)GOTO 9
C      IF(K.LE.6)GOTO 10
C      GOTO 11
C      PERFORM LEAST-SQUARES FIT TO DATA WITH A STRAIGHT LINE
C      9
C      ALY=LOG(TY)
C      SUMX=SUMX+DX/TY
C      SUMX2=SUMX2+DX*DX/TY
C      SUMY=SUMY+ALY/TY
C      SUMXY=SUMXY+DX*ALY/TY
C      SUMW=SUMW+1./TY
C      K=K+1
C      10 CONTINUE
C      IF LESS THAN 4 POINTS ARE INVOLVED RETURN AND USE DEFAULT VALUES
C      11 IF(K.LT.3)RETURN
C      GET DETERMINANT. IF DETERMINANT IS EQUAL TO ZERO, RETURN WITHOUT
C      FURTHER CALCULATION.
C

```

```
11 IF(K.LT.3)RETURN
C
C GET DETERMINANT. IF DETERMINANT IS EQUAL TO ZERO, RETURN WITHOUT
C FURTHER CALCULATION.
C
C   DEL =SUMM*SUNX2-SUMX*SUMX
C   IF(DEQ.0)RETURN
C   A=(SUMX2*SUMY-SUMX*SUNXY)/DEL
C SHORT-TERM TAIL AMPLITUDE
C   A=DEXP(A)/PKHT
C SHORT-TERM TAIL SLOPE
C   B=(SUMM*SUNXY-SUMX*SUNY)/DEL
C
C   RETURN
C   END
```

\$large

SUBROUTINE DCHDAT(Nref,std,fname)

DCHDAT

=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCEM, Inc.
3099B Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : DCHDAT
Revision : 1.0
Author : Russell Fleming
Date : 4/1/87
Language : Microsoft Fortran Ver. 4.01

Abstract : Gets temporary data from dBASE input for dcheck

Arguments :

Routines
Called : None

Libraries
Used : None

Revision History (Include date, revision I.D., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

variable initialization declarations

Dimension std(20),nref(2)
CHARACTER FNAME*30
real tempnref(2)

```

C *****
C      format declarations
C *****
C
10      format(f10.3)
20      format(A1)
30      format(A30)
40      format(i12)
50      format(f8.3)
C *****
C
      OPEN(7,FILE='DCHECK.PRM',IOSTAT=istat)
      IF(istat.ne.0)goto 90
      READ(7,40)nref(1)
      READ(7,40)nref(2)
      READ(7,50)std(1)
      READ(7,50)std(2)
      READ(7,30)fname
      go to 99
90      WRITE(*,*)' Error Opening Parameter File,Press Enter to Continue'
      pause','
      close(7)
      stop
99      CLOSE(7)
      RETURN
      END

```

Slarge

```

      subroutine decay(wper,pd,time,indx)
C*****
C
C          DECAP
C          =====
C*****
C
C      Developed by : Safeguards Technology Program
C                   Nuclear Chemistry Division
C                   Lawrence Livermore National Laboratory
C                   Livermore, CA. 94550
C
C      This work supported by LLNL PO# 9107605
C*****
C
C      Title       : DECAP
C      Revision    : 1.0
C      Author      : Wayne D. Ruhter and Russell Fleming
C      Date        : 11/9/87
C      Language    : Microsoft Fortran Ver. 4.01
C*****
C
C      Abstract    : Performs decay correction on isotopic abundances.
C*****
C
C      Arguments  : wper,pd,time,indx
C*****
C
C      Routines
C      Called    : None
C
C      Libraries
C      Used      : None
C*****
C
C      Revision History ( Include date,revision I.D., and details )
C*****
C
C      COMMON and variable declarations
C*****
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C      dimension wper(6),time(6)
C*****
C
C      variable initialization declarations
C*****
C
C      wper(indx)=wper(indx)*dexp(-(0.693*pd*1440./time(indx)))
C
C      return
C      end

```


\$large

subroutine diaea

DIAEA
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : DIAEA
Revision : 1.0
Author : Russell Fleming
Date : 11/9/87
Language : Microsoft Fortran Ver. 4.01

Abstract :

Saves IAEA Analysis variables to tmp.prm file for dBASE database

Arguments :

Routines
Called : None
Libraries
Used : None

Revision History (Include date, revision I.O., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

variable initialization declarations

\$include: 'lreadb.cmn'

character filen*10,temp*150,specname*30

C

```

filen='iaea.dbf'
write(temp,10)cp238,cpe238,cp239,cpe239,cp240,
+ cpe240,cp241,cpe241,cp242,cpe242,
+ cpow,cpowe,ce240,ce240e,specname

```

```

10 format(14(f10.4),a30)
open(unit=4,file='tmp.prm',iostat=iostat)
if (iostat.ne.0)then
write(*,*)'Error opening temporary file TMP.PRM'
pause 'Please press Enter to continue'
goto 99
endif
write(4,20)temp
20 format(a180)
99 return
end

```

\$large

```

subroutine dsave
.....
                                DSAVE
                                =====
.....

Developed for : Safeguards Technology Program
                Nuclear Chemistry Division
                Lawrence Livermore National Laboratory
                Livermore, CA. 94550

Developed by :  COMPUCHEN, Inc.
                30998 Huntwood Avenue, Suite 107
                Hayward, CA. 94544
                (415)489-6514

This work supported by LLNL POW 9107605
.....

Title       : DSAVE
Revision    : 1.0
Author      : Russell Fleming
Date       : 11/9/87
Language    : Microsoft Fortran Ver. 4.01
.....

Abstract    : Save dcheck variables to dcheck.dbf dbase file
.....

Arguments   :

.....

Routines
Called      : None

Libraries
Used        : None
.....

Revision History ( Include date, revision I.D., and details )
.....

COMMON and variable declarations
.....

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
.....

variable initialization declarations
.....

$INCLUDE: 'Top.cmn'

common /dchk/ nref, std, livtim, crate, zero, gain, wfuhn, sfuhn,
+             shpc, parea, spare, outlo, fnum

```

```

dimension frames(127,2), flens(127,2), finfo(4), ndata(127),
+ cdata(127), nref(2), std(20), shapc(2),
+ parae(2), sparae(2), wfuwm(2), wfuwm(2)

character frames*11, cdata*127, fname*30, flen*45
real*4 ndata
integer*2 flens, lunit, nrec, li, strlen
integer*4 finfo, livitm

logical*2 puglg
logical debug

debug=.false.

if(strlen(drivepath)
if (debug) write(*,*)'Length of "',drivepath,'" = ',li
file=drivepath(1:li)///'check.dbr'
if (debug) write(*,*)'std(1)="'std(1)'"', file, '"'
if (debug) pause

lunit=3
do 20 i=1,127
frames(i,1)=i
frames(i,2)=i

call dbopen(file, lunit, frames, flen, finfo)

cdata(1) = fname
if (debug) write(*,*)'fname=',fname
ndata(1) = float(nref(1))
if (debug) write(*,*)'nref(1)='',nref(1)
ndata(2) = angl(std(1))
if (debug) write(*,*)'std(1)='',std(1)
ndata(3) = float(nref(2))
if (debug) write(*,*)'nref(2)='',nref(2)
ndata(4) = angl(std(2))
if (debug) write(*,*)'std(2)='',std(2)
ndata(5) = float(livitm)
if (debug) write(*,*)'livitm='',livitm
ndata(6) = angl(std(2))
if (debug) write(*,*)'std(2)='',std(2)
ndata(7) = angl(zero)
if (debug) write(*,*)'zero='',zero
ndata(8) = angl(gain)
if (debug) write(*,*)'gain='',gain
ndata(9) = angl(wfuwm(1))
if (debug) write(*,*)'wfuwm(1)='',wfuwm(1)
ndata(10) = angl(wfuwm(1))
if (debug) write(*,*)'wfuwm(1)='',wfuwm(1)
ndata(11) = angl(shapc(2))
if (debug) write(*,*)'shapc(2)='',shapc(2)
ndata(12) = angl(shapc(1))
if (debug) write(*,*)'shapc(1)='',shapc(1)
ndata(13) = angl(parae(1))
if (debug) write(*,*)'parae(1)='',parae(1)
ndata(14) = angl(sparae(1))
if (debug) write(*,*)'sparae(1)='',sparae(1)
ndata(15) = angl(wfuwm(2))
if (debug) write(*,*)'wfuwm(2)='',wfuwm(2)
ndata(16) = angl(sfuwm(2))
if (debug) write(*,*)'sfuwm(2)='',sfuwm(2)
ndata(17) = angl(parae(2))
if (debug) write(*,*)'parae(2)='',parae(2)

```

```
      if (debug) write(*,*)'sparea(2)=',sparea(2)
      if (puflg) then
        cdata(2) = 'I'
      else
        cdata(2) = 'F'
      end if

      nrec = finfo(4)+1
      call dbwrite(filcn,iunit,fnames,flens,finfo,cdata,ndata,nrec)
99    return
      end
```

\$large

SUBROUTINE GFIT2(L,ALFA,EXP1,EXP2,PKPOS,PKHT,ERR,Y)

GFIT2
=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL PCW 9107605

Title : GFIT2
Revision : 1.0
Author : Wayne D. Ruher
: Revised for the IBM PC by Russell Fleming
COMPUchem, Inc.
3099B Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514
Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :
This subroutine fits the upper half of a peak with a gaussian function after stripping out the tailing contribution. The subroutine returns the peak position, height, and width. An error condition is returned if no valid peak exists within the region indicated or if the statistics are poor. The calling program must then take appropriate action.

Arguments :
L = LOCATION WITHIN "Y" ARRAY OF PEAK POSITION
ALFA = PEAK WIDTH PARAMETER; AN APPROXIMATE VALUE SHOULD BE SENT TO THIS ROUTINE. A CALCULATED VALUE IS RETURNED.
EXP1 = SHORT TERM TAIL AMPLITUDE
EXP2 = SHORT TERM TAIL SLOPE
PKPOS = POSITION OF PEAK CALCULATED BY THIS ROUTINE
PKHT = PEAK HEIGHT CALCULATED BY THIS ROUTINE
ERR = ERROR FLAG; SET TO -1 IF AN ERROR CONDITION EXISTS; OTHERWISE THE ERROR ON THE PEAK POSITION
Y = DATA ARRAY

Routines
Called : None
Libraries
Used : None

Revision History (Include date, revision I.D., and details) *

Date: 25-MAR-80

Update: 23-JUL-82

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

DIMENSION Y(210)

DOUBLE PRECISION SY,SYV,SKY,SKXY,SK2,SKX

DOUBLE PRECISION XPX11,XPX22,XPX33,A0,A1,Y1

variable initialization declarations *

IN CASE OF EARLY RETURN

PKPOS = L

CLEAR ERROR FLAG

ERR = 0

C EXPECTED CHANNEL HALF-WIDTH

I = SORT(.693/(-1.0*ALFA))

INITIALIZE SUMS

SY = 0.0

SYV = 0.0

SKY = 0.0

SKXY = 0.0

SKX = 0.0

SK2 = 0.0

format declarations *

SEARCH FOR THE CHANNEL WITH DATA LESS THAN 1/2 THE
MAXIMUM HEIGHT OF THE PEAK.

Y2=0.5*Y(L)

M=1

DO 10 J=1,12

IF(Y(L+M)-L.T.Y2)GOTO 11

M=M+1

CONTINUE

TEST TO SEE IF A CHANNEL WAS FOUND AND HOW FAR AWAY
IF A CHANNEL WAS NOT FOUND OR IT IS LESS THAN 2 CHANNELS
AWAY FROM THE MAXIMUM CHANNEL RETURN WITH AN ERROR CONDITION

11 IF((M-GE.1-1).AND.(M-LT.12))GOTO 12

ERR = -1

RETURN

```

C
12  C=M*(M+1)/3.
    Y1 = Y(L+1)
    Y2 = Y(L-1)
    TPOS = .5 * (Y2-Y1) / (Y1 - 2. * Y(L) + Y2)
    Y0 = Y(L)

C
C    STRIP OUT THE SHORT-TERM TAIL CONTRIBUTION TO THE PEAK
C
C    I=(-1.0*M)
100  J=L+1
    XL=I
    DLX = XL - TPOS
    Y1 = Y(J)
    IF (DLX) 20,27,27
20  Y1=Y1-Y0 * EXP1*DEXP(EXP2 * DLX)*(1.-DEXP(.4*ALFA*DLX*DLX))
27  IF(Y1.LE.0) Y1 = .1

C
C    TAKE THE NATURAL LOGARITHM OF THE NET "Y" DATA; THE RESULT
C    SHOULD BE A PARABOLA. FIT THE RESULTING VALUES TO A PARABOLA
C    TO GET THE PEAK HEIGHT, POSITION, AND WIDTH:
C
    Y1=DLOG(Y1)
    SY=SY+Y1
    SYY=SYY+Y1**2
    SXY=SXY+XL*Y1
    SXXY=SXXY+(XL**2-C)*Y1
    SXX =SXX +(XL**2-C)**2
    SX2 =SX2 +XL*XL
99  I=I+1
    IF(1.LE.M)GOTO 100

C
C    COMPUTE THE DIAGONAL ELEMENTS OF X'X
C
    XPX11=2*M+1.
    XPX22=XPX11*C
    XPX33=XPX22*(4*M*(M+1)-3)/15.

C
C    COMPUTE LEAST SQUARES ESTIMATES OF COEFFICIENTS
C
    A0=SY/XPX11
    A1=SXY/XPX22
    ALFA=SXXY/XPX33

C
C    IS THE PARABOLA NEARLY DEGENERATE; IF SO, RETURN WITH ERROR
C
    IF(ABS(A0).GE.1.E-06)GOTO 4
    ERR=.1
    RETURN

C
C    COMPUTE THE COORDINATES OF THE VERTEX
C
4    PKPOS=-0.5*A1/ALFA
    PKHT=A0+0.5*A1*PKPOS-ALFA*C
    ERR=SYY-2.*A0*SY-2.*A1*SXY-2.*ALFA*SXXY + (2*M+1)*A0**2
1  +A1**2*SX2 + ALFA**2*SXX
    IF(M.GT.1)ERR=ERR/(2*(M-1))
    PKPOS=PKPOS+L
    PKHT=DEXP(PKHT)

C
C    CALCULATE THE ERROR ON ALFA AND RETURN IT
C
    ERR = ERR * SXX/XPX33**2
    IF (ERR .LT. 0.)ERR = 0.
    RETURN
END

```

\$large

SUBROUTINE GETD

GETD

====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUchem, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : GETD
Revision : 1.1
Author : Russell Fleming
Date : 11/16/87
Language : Microsoft Fortran Ver. 4.01

Abstract : Gets temporary data from DBASII input for analysis

Arguments :

Routines
Called : None
Libraries
Used : None

Revision History (include date, revision I.D., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

\$INCLUDE: 'iaea.cmn'

CHARACTER jnk*1
logical debug

variable initialization declarations

```

C .....
C
C      Debug=.false.
C .....
C      format declarations
C .....
C
10      format(F10.3)
20      format(A1)
30      format(A30)
35      format(A)
40      format(I10)
50      format(F8.3)
C .....
C
C      if (debug) then
C          write(*,*)char(7)
C          write(*,*)'In GETDATA'
C          pause
C          endif
C
C      ...initialize Pu and Am dates
C
C          amdate = ' '
C          pudate = ' '
C          OPEN(7,FILE='OUT_IA.PRM',Iostat=istat)
C          IF(istat.ne.0)goto 90
C          read(7,20)jnk
C          if (debug) write(*,*)'jnk=' ,jnk
C          READ(7,40)nref(1)
C          if (debug) write(*,*)'nref(1)=',nref(1)
C          READ(7,40)nref(2)
C          if (debug) write(*,*) 'nref(2)=',nref(2)
C          READ(7,10)tp238
C          if (debug) write(*,*)'p238=',tp238
C          READ(7,10)tp239
C          if (debug) write(*,*)'p239=',tp239
C          READ(7,10)tp240
C          if (debug) write(*,*)'p240=',tp240
C          READ(7,10)tp241
C          if (debug) write(*,*)'p241=',tp241
C          READ(7,10)tp242
C          if (debug) write(*,*)'p242=',tp242
C          READ(7,10)ta241
C          if (debug) write(*,*)'Am241=',tm241
C          read(7,10)tp238
C          if (debug) write(*,*)'pe238=',tpe238
C          read(7,10)tp239
C          if (debug) write(*,*)'pe239=',tpe239
C          read(7,10)tp240
C          if (debug) write(*,*)'pe240=',tpe240
C          read(7,10)tp241
C          if (debug) write(*,*)'pe241=',tpe241
C          READ(7,10)tp242
C          if (debug) write(*,*)'pe242=',tpe242
C          READ(7,10)ta241
C          if (debug) write(*,*)'Am241=',tpe241
C          READ(7,50)coltim
C          if (debug) write(*,*)'Coltim=',coltim
C          READ(7,30)fname
C          if (debug) write(*,*)'fname=',fname
C          READ(7,30)rname
C          if (debug) write(*,*)'Rname=',rname
C          READ(7,20)printer

```

```
      READ(7,35)rep
      if (debug) write(*,*)'rep=',rep
      READ(7,35)specdate
      if (debug) write(*,*)'Specdate=',specdate,':'
      READ(7,35)spectime
      if (debug) write(*,*)'spectime=',spectime,':'
      READ(7,35)Pudate
      if (debug) write(*,*)'Pudate=',Pudate,':'
      READ(7,35)Amdate
      if (debug) write(*,*)'Amdate=',Amdate,':'
      READ(7,35)Sample
      if (debug) write(*,*)'Sample=',sample,':'
      if (debug) pause
      exflag=' '
      GOTO 99

90  WRITE(*,*)' Error Opening Parameter File,Press Enter to Continue'
      read(*,30)jnk
      EXFLAG='X'
99  CLOSE(7)
      RETURN
      END
```

\$large

DOUBLE PRECISION FUNCTION DPM(XX,L)

DPM

=====

Developed by : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

This work supported by LLNL PO# 9107605

Title : DPM
 Revision : 1.0
 Author : Wayne D. Ruhter
 : Revised for the IBM PC by Russell Fleming
 : COMPUCHEM, Inc.
 : 30998 Huntwood Avenue, Suite 107
 : Hayward, CA. 94544
 : (415)489-6514

Date : 10/15/86
 Language : Microsoft Fortran Ver. 4.01

Abstract :

Arguments : XX,L

Routines
 Called : None

Libraries
 Used : None

Revision History (Include date, revision I.D., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /MAIN/ E(9),

EFF(210),

ALPHA(6,6),

ANS(6),

ERR(7),

YFIT(10),

SIGMAA(7)

variable initialization declarations

xtt=.208
 ENL208 = dLOG(xtt)
 ENLXX = dLOG(XX)

DPM=ANS(L)+ANS(3)*(ENLXX-ENL208)
 1 + ANS(4)*(ENLXX**2-ENL208**2)
 2 + ANS(5)*(ENLXX**3-ENL208**3)

DPM=dEXP(DPM)

RETURN
 END

\$Large
PROGRAM Master

Master
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA, 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : Master
Revision : 1.0
Author : Russell Fleming
Date : 4/28/87
Language : Microsoft Fortran Ver. 4.01

Abstract : This program will call Iaeapu or Dcheck.

Arguments :

Routines
Called : IAEAPU
DCHECK

Libraries
Used : Fortran.Lib

Revision History (Include date, revision I.D., and details) *

COMMON and variable declarations *

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

\$INCLUDE: 'Top.cmn'

character action,junk
format(1a)

c ---

```
Open(9,file='SWITCH.PRM',iostat=istat)
if (istat.ne.0) then
  write(*,*)
  write(*,*)' Error opening SWITCH.PRM, error=',istat
  write(*,*)' Please Press Enter to Continue.'
  read(*,20)junk
  stop
end if
read(9,10)action
read(9,10)drivepath
format(a)
close(9)
if (action.eq.'1') then
  call Iaeapu
else
  if (action.eq.'D') then
    call dcheck
  else
    call cicero
  end if
end if
end
```

10

---->

\$large

SUBROUTINE MATINV(ARRAY,MX,NORDR,DET)

MATINV

=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA 94550

This work supported by LLNL POW 9107605

Title : MATINV
Revision : 1.00
Author : Wayne D. Ruhter, modified by
Russell Fleming and William Buckley
for MS/PC-DOS
Date : 12 May 1986
Language : MS FORTRAN Ver. 4.01 (MS/PC-DOS)

Abstract:

This subroutine was adapted from the matrix inversion subroutine given in the book, "Data Reduction and Error Analysis for the Physical Sciences" by Philip R. Bevington. This subroutine inverts a square symmetric matrix and calculates its determinant. The input variables are ARRAY and NORDR, and the output variable is DET. The inverted matrix is returned in ARRAY. MX must be equal to the size of the array in the calling program.

DICTIONARY OF VARIABLES

ARRAY = MATRIX WHOSE INVERSE IS TO BE EVALUATED
MX = ARRAY DIMENSION
NORDR = DEGREE OF THE MATRIX
DET = VALUE OF THE DETERMINANT

Compiler Directives -- Metacommands

\$NOFLOATCALLS
\$NOLIST

Routines called : NONE

Called by : IAEAPU

Libraries used : 8087.LIB

COMMON and local variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

Variable and parameter initialization

DIMENSION ARRAY(MX,MX),IK(50),JK(50)

Format declarations

DET = 1.

DO 100 K = 1,NORDR

FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX

AMAX = 0.

DO 30 I = K,NORDR

DO 30 J = K,NORDR

IF(ABS(AMAX)-ABS(ARRAY(I,J)))24,24,30

AMAX = ARRAY(I,J)

IK(K) = I

JK(K) = J

CONTINUE

INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)

IF(AMAX)41,32,41

DET = 0.

GOTO 140

I = IK(K)

IF(I-K)21,51,43

DO 50 J = 1,NORDR

ARRAY(I,J) = -ARRAY(I,J)

J = JK(K)

IF(J-K) 21,61,53

DO 60 I = 1,NORDR

SAVE = ARRAY(I,K)

ARRAY(I,K) = -ARRAY(I,J)

ARRAY(I,J) = -SAVE

ACCUMULATE ELEMENTS OF INVERSE MATRIX

DO 70 I = 1,NORDR

IF(I-K)63,70,63

ARRAY(I,K) = -ARRAY(I,K)/AMAX

CONTINUE

DO 80 I = 1,NORDR

DO 80 J = 1,NORDR

IF(I-K)74,80,74

IF(J-K)75,80,75

ARRAY(I,J) = ARRAY(I,J) + ARRAY(I,K) * ARRAY(K,J)

CONTINUE

DO 90 J = 1,NORDR

IF(J-K)83,90,83

ARRAY(K,J) = ARRAY(K,J)/AMAX

CONTINUE

ARRAY(K,K) = 1. / AMAX

CONTINUE

```
90      CONTINUE
      ARRAY(K,K) = 1. / AMAX
100     CONTINUE
C
C      RESTORE ORDERING OF MATRIX
C
101     DO 130 L = 1, NORDER
      K = NORDER + L + 1
      J = IK(K)
      IF (J - K) 111, 111, 105
105     DO 110 I = 1, NORDER
      SAVE = ARRAY(I,K)
      ARRAY(I,K) = -ARRAY(I,J)
110     ARRAY(I,J) = SAVE
111     I = JK(K)
      IF (I - K) 130, 130, 113
113     DO 120 J = 1, NORDER
      SAVE = ARRAY(K,J)
      ARRAY(K,J) = -ARRAY(I,J)
120     ARRAY(I,J) = SAVE
130     CONTINUE
140     RETURN
      END
```

\$large

SUBROUTINE MAXVAL (NS,NE,JMAX,YMAX,Y)

```

      MAXVAL
      =====

```

```

      Developed by : Safeguards Technology Program
                   : Nuclear Chemistry Division
                   : Lawrence Livermore National Laboratory
                   : Livermore, CA. 94550

```

```

      This work supported by LLNL PD# 9107605

```

```

      Title       : MAXVAL
      Revision    : 1.0
      Author      : R. Gunnink
                  : Revised for the IBM PC by Russell Fleming
      Date       : 10/15/86
      Language    : Microsoft Fortran Ver. 4.01

```

```

      Abstract :
      This routine returns the position and value of the maximum value
      in a group of data points from NS to NE inclusive.

```

Arguments :

```

      INPUT
      NS      starting position for group in Y array
      NE      ending position for group in Y array
      Y       data array

      OUTPUT
      JPK     position in array Y of maximum data value
      YMAX    maximum data value in group set by NS and NE

```

```

      Routines
      Called   : None

```

```

      Libraries
      Used     : None

```

```

      Revision History ( Include date,revision I.D., and details )

```

```

      Version: 2.1
      Author:  R. Gunnink
      Date:    19-JAN-78

```

```

      COMMON and variable declarations

```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```

      DIMENSION Y(210)

```

```

      C
      C *****
      C      variable initialization declarations
      C *****
      C
      C      YMAX = 0.0
      C *****
      C
      C      DO 2 I = NS,NE-1
      C          IF (Y(I) - YMAX) 2,2,4
      C          YMAX = Y(I)
      C          JMAX = I
      C      2    CONTINUE
      C      RETURN
      C      END

```

\$large

SUBROUTINE MORE(ILN,prn)

```

      MORE
      =====

```

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

```

Title       : More
Revision    : 1.0
Author      : Russell Fleming
Date        : 12/8/86
Language    : Microsoft Fortran V. 4.01

```

Abstract : This will pause after the screen is filled with text.

Arguments : Iln, Prn

```

Routines
Called  :
Libraries
Used    :

```

Revision History (Include date, revision I.D., and details) *

variable initialization declarations *

```
CHARACTER*1 JUNK,prn
```

format declarations *

```
FORMAT(/,' Press Enter to continue ... ',%)
format(1a)
```

```

C*****
C

```

```

      if(prn.eq.'S')then
        ILN=ILN+1
        IF (ILN.GT.20) THEN
          ILN=0
          WRITE(*,10)
          READ(*,20)JUNK
          WRITE(*,*)
        END IF
      end if
      RETURN
      END

```

\$large

SUBROUTINE NCNTS (N1,N2,M1,M2,ND,S1,S2,ER,AVEBG,Y1,Y2)

NCNTS

=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL POW 9107605

Title : NCNTS
Revision : 1.0
Author : Wayne D. Ruhter
: Revised for the IBM PC by Russell Fleming
Date : 11/9/87
Language : Microsoft Fortran Version 4.01

Abstract :
THIS ROUTINE IS SIMILAR TO "NCIS" ROUTINE EXCEPT THAT IT RETURNS
AN AVERAGE BACKGROUND AND THE ERROR ASSOCIATED WITH THE TOTAL NET
COUNT.

Arguments : N1,N2,M1,M2,ND,S1,S2,ER,AVEBG,Y1,Y2

Routines
Called : AVE, BKGD
Libraries
Used : None

Revision History (Include date, revision 1.0., and details) *

NCNTS VERSION 0 19-JAN-78 AT 12:30

COMMON and variable declarations *

IMPLICIT DOUBLE PRECISION (A-H,O-Z).
DIMENSION Y1(2),Y2(2)

variable initialization declarations *

```

C*****
C      format declarations
C*****
C
C*****
C
C      CALL AVE (N1,N2,S1,AV1,Y1)
C      CALL AVE (N1,N2,S2,AV2,Y1)
C      S1 = S1 + S2
C      CALL BKGD (N2+1,M1-1,AV1,AV2,S2,ND,Y1,Y2)
C      CHANS = ND
C      XPTS = M2 + M2 - M1 - M1 + 1
C      AVEBG = 0.5 * (AV1 + AV2)
C      IF (S2 .LT. 0.0) S2 = 0.0
C      ER = SQRT (S2 + AVEBG * CHANS * (1. + CHANS / XPTS))
C      RETURN
C      END

```

---->

Slarge

SUBROUTINE PUFIT(NTERMS,NDPTS,CHISQR)

PUFITY

=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL POW 9107605

Title : PUFITY
Revision : 1.0
Author : Wayne D. Ruhter
Revised for the IBM PC by Russell Fleming
COMPUCEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :

Arguments : NTERMS,NDPTS,CHISQR

Routines
Called : None

Libraries
Used : None

Revision History (Include date,revision I.D., and details) *

COMMON and variable declarations *

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON E, EFF, ALPHA, A, ERR, YFIT, SIGMAA
      COMMON /MAIN/ E(9),
1         EFF(210),
1         ALPHA(6,6),
1         A(6),
1         ERR(7),
1         YFIT(10),
1         SIGMAA(7)

```

```

      C      DIMENSION E(9),EFF(210),ERR(7),SIGMAA(7),YFIT(10)
      C      DOUBLE PRECISION WEIGHT(10),P(10,6),BETA(6),Y(10)

```

C variable initialization declarations *

C ENL208 = LOG(.208)

C format declarations *

906 FORMAT(' MATRIX IS SINGULAR')

```

      C
      DO 10 J=1,NDPTS
      DO 10 K=1,NTERMS
      P(J,K)=0.

```

```

      P(1,2)=1.
      P(2,1)=1.
      P(3,2)=1.
      P(4,1)=1.
      P(5,1)=1.
      P(6,2)=1.
      P(7,2)=1.
      DO 20 I=1,NDPTS
      SIGMAA(I)=ERR(I)/100.
      WEIGHT(I) = 1./SIGMAA(I)**2
      Y(I)=LOG(EFF(I))
      P(1,3)=E(I) * ENL208
      P(1,4)=E(I)**2 * ENL208**2
      P(1,5)=E(I)**3 * ENL208**3

```

40 CONTINUE

C ACCUMULATE MATRICES ALPHA AND BETA

```

      C
      DO 54 J=1,NTERMS
      BETA(J)=0.
      DO 54 K=1,NTERMS
      ALPHA(J,K)=0.
      DO 66 I=1,NDPTS
      BETA(J)=BETA(J)+P(I,J)*Y(I)*WEIGHT(I)
      DO 66 K=J,NTERMS
      ALPHA(J,K)=ALPHA(J,K)+P(I,J)*P(I,K)*WEIGHT(I)
      ALPHA(K,J)=ALPHA(J,K)

```

C INVERT CURVATURE MATRIX

```

      C
      DO 95 J=1,NTERMS
      A(J)=0.
      DO 97 I=1,NDPTS
      YFIT(I)=0.
      CALL MATINV(ALPHA,6,NTERMS,DET)
      IF(DET)11,103,111
      CHISQR=0.
      GOTO 170

```

C CALCULATE COEFFICIENTS,FIT, AND CHI-SQUARE

```

111      DO 113 J=1, NTERMS
113          DO 113 K=1, NTERMS
              A(J)=A(J)+BETA(K)*ALPHA(J,K)
          DO 115 I=1, NDPTS
              YFIT(I)=A(2)
              IF(1.EQ.2)YFIT(1)=A(1)
              IF(1.EQ.4)YFIT(1)=A(1)
              IF(1.EQ.5)YFIT(1)=A(1)
          DO 115 J=3, NTERMS
              YFIT(1)=YFIT(1) + A(J) * P(I,J)
121      CHISO=G.
          DO 123 I =1, NDPTS
              YFIT(I)=DEXP(YFIT(1))
              SIGMAA(I)=ERR(I)*EFF(I)/100.
              WEIGHT(I)=1./SIGMAA(I)**2
              DIF=EFF(I)-YFIT(I)
              PCTDIF=(DIF/EFF(I))*100.
              X=DEXP(E(I))
123      CHISO=CHISO+DIF**2*WEIGHT(I)
          FREE = NDPTS*NTERMS
          IF(FREE.EQ.0)GOTO 155
          CHISQ=CHISO/FREE

C          CALCULATE REMAINDER OF OUTPUT
C
155      DO 156 J=1, NTERMS
156          SIGMAA(J)=SQRT(ALPHA(J,J))
          RETURN
170      write(*,906)
          RETURN
          END

```


\$Large

SUBROUTINE RDBIN(isum,DUM,NDEV)

RDBIN

=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : RDBIN
Revision : 1.0
Author : Russell Fleming
Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :
This program reads binary data files.

Arguments : NST,NDPTS,DUM,NDEV

Routines
Called : None

Libraries
Used : None

Revision History (Include date,revision I.D., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INTEGER*4 DUM(8000)

variable initialization declarations

```

*****
C      format declarations
C      *****
C
C
C      isum=1
C      read(ndev,iostat=iflag)dum(isum)
5      if ((iflag.lt.0).or.(isum.ge.4100)) goto 10
C      isum=isum+1
C      READ(NDEV,iostat=iflag)DUM(isum)
10     goto 5
C      CLOSE(NDEV)
C      RETURN
C      END

```

\$large

SUBROUTINE RDSK(NST,NDPTS,DUM,NDEV)

RDSK

=====

Developed by : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

This work supported by LLNL PC# 9107605

Title : RDSK
 Revision : 1.0
 Author : Wayne D. Ruhter
 : Revised for the IBM PC by Russell Fleming
 : COMPUCHEM, Inc.
 : 30998 Huntwood Avenue, Suite 107
 : Hayward, CA. 94544
 : (415)489-6514
 Date : 10/15/86
 Language : Microsoft Fortran Ver. 4.01

Abstract :
 This program reads ASCII data files in 6A1 format.

Arguments : NST,NDPTS,DUM,NDEV

Routines
 Called : None

Libraries
 Used : None

Revision History (include date, revision I.D., and details)

Version 1.0 18-MAY-82

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 LOGICAL*2 DATA(1200)
 character*10 DD

DIMENSION DUM(0:4100)

```

C *****
C variable initialization declarations
C *****
C
C *****
C format declarations
C *****
C
200 FORMAT(A10)
202 FORMAT(F10.0)
C *****
C
C
10 DO 10 I=1,NST
   READ(NDEV,200)DD
   IEND=NST + NDPTS
   I = 0
   DO 50 J=IEND,IEND
     READ(NDEV,200)DD
     write(*,*)'DD=',dd
     I = I + 1
     read(DD,202)DUM(I)
     write(*,*)DUM(I)
   C
   50 continue
     CLOSE(NDEV)
     RETURN
     END

```

---->

\$large

SUBROUTINE SPPOW(PCT,RAT,POW,POME,P242FG,P242,PE242,E240,E240E)

SPPOW
=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL PO# 9107605

Title : SPPOW
Revision : 1.0
Author : Wayne D. Ruhter
Revised for the IBM PC by Russell Fleming
COMPUCHER, Inc.
3099B Hunter-d Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :
This subroutine calculates the specific power and Pu-240 effective for the sample based on the isotopic abundances for the plutonium isotopes and Am-241. It also calculates the errors on these quantities.

Arguments : PCT,RAT,POW,POME,P242FG,P242,PE242,E240,E240E

Routines : None
Called :

Libraries : None
Used :

Revision History (Include date,revision I.D., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

variable initialization declarations

COMMON /MAIN/ X(9),
Y(210),
ALPHA(6,6),
SUM(6),
ERROR(7),
YFIT(10),
SIGMA(7)

logical*2 P242FG

DIMENSION PCT(6),PCTE(6),SPHT(6),E240C(3),RAT(4)

Specific power values for each of the isotopes

DATA SPHT/567.16,1.929,7.098,3.339,.1146,114.23/
DATA E240C/2.49,1.,0./

Calculate the specific power for the sample

POW = 0.
POME = 0.
DO 10 J = 1,6
SPW = .01 * SPHT(J) * PCT(J)
POW = POW + SPW

Calculate error on specific power. This calculation ignores the error contribution from Pu-242, because it is negligible and the calculation is greatly simplified by doing so.

L = 0
DO 20 J = 1,6
IF (J.EQ.2 .OR. J.EQ.5) GO TO 20
L = L + 1
ATERM = POW - SPHT(J)
IF (J.EQ.6) ATERM = SPHT(J)
M = 0
DO 19 K = 1,6
IF (K.EQ.2 .OR. K.EQ.5) GO TO 20
M = M + 1
BTERM = POW - SPHT(K)
IF (K.EQ.6) BTERM = SPHT(K)
ADD = ATERM*BTERM*PCT(J)*PCT(K)*ALPHA(L,M)/(RAT(L)*RAT(M))
POME = POME + ADD

continue
CONTINUE
POME = SQRT(POME)/100.

Calculate the Pu-240 effective

E240 = (2.49 * PCT(1) + PCT(3) + 1.57 * PCT(5)) / 100.

Calculate error on Pu-240 effective for situation where Pu-242 abundance is entered by operator.

E240E = 0.
L = 0
EADJ = 0.
DO 30 J = 1,3
ATERM = E240C(J) - E240 + 1.57*P242*.01
IF (P242FG .AND. J.GE.2)EADJ = 1.57*P242*.01
L = L + 1
IF (J.EQ.2)L = L + 1
M = 0
FADJ = 0.
DO 30 K = 1,3

```
      BTERM = E240C(K) * E240 + 1.57*P242*.01
      IF (P242FG .AND. K.GE.2)FADJ= 1.57*P242*.01
      M = M + 1
      IF (K.EQ.2)M = M + 1
      ADD = (ATERM*PCT(L)*.01+EADJ)*(BTERM*PCT(M)*.01+FADJ)
1      *ALPHA(J,K)/(RAT(J)*RAT(K))
      E240E = E240E + ADD
30     CONTINUE
      IF (.NOT.P242FG)GO TO 35
      E240E = E240E + 2.465*(PE242*.01)**2
35     E240E = 100. * SORT(E240E)
      E240 = 100. * E240
      RETURN
      END
```

\$large

SUBROUTINE STRIP1 (NS,NE,PKPOS,PKHT,ALFA,EXP1,EXP2,WL,Y)

```

      STRIP1
      =====

```

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL POF 9107605

```

      Title   : STRIP1
      Revision : 1.0
      Author  : R. Gunnink
               Revised for the IBM PC by Russell Fleming
               COMPUCEM, Inc.
               3099B Huntwood Avenue, Suite 107
               Hayward, CA. 94544
               (415)489-6514

      Date    : 10/15/86
      Language : Microsoft Fortran Ver. 4.01

```

Abstract :
This subroutine is used to strip a peak or part of a peak from
a continuum. STRIP1 can be used on high-energy x-rays.

Arguments :

```

INPUT
NS = Starting position in data array from which peak is to be
    stripped
NE = Ending position in data array
PKPOS = Position in array of peak to be stripped
PKHT = Peak height
ALFA = Full-width-half-maximum parameter of peak to be stripped
EXP1 = Short-term tail amplitude of peak
EXP2 = Short-term tail slope of peak
WL = Lorentzian width of peak, if peak is an x-ray
Y = Data array containing data from which peak is stripped

```

```

Routines
Called : None

Libraries
Used : None

```

Revision History (include date, revision I.D., and details)

```

C Version: 2.1
C Author: R. Gunnink
C Date: 20-JAN-78
C
C *****
C COMMON and variable declarations
C *****
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C DIMENSION Y(210)
C
C *****
C variable initialization declarations
C *****
C
C *****
C format declarations
C *****
C *****
C
C IF (WL)1,2,1
C1 = SQRT (-ALFA)
C2 = WL * CX
C3 = 1. - .5642 * GAMA
C4 = .15915 * GAMA
C5 = .25 * GAMA * GAMA
C6 = .6356 * GAMA
RT = C1 + C3
2 DO 10 I = NS,NE
  XI = I
  DX = XI - PKPOS
  DDX = DX * DX
  ADX = ALFA * DDX
  FXX = EXP (ADX)
  FX = FXX
  IF (DX) 3,4,4
  EXPN2 = EXP (EXP2 * DX)
  FXX = FXX + EXP1 * EXPN2 * (1. - EXP (.4 * ADX))
  DX = -DX
  IF (WL) 6,9,6
  CC = CX * DX
  CALL BWL (CC)
  FXX = FXX * FX + (C1 - C2 * ADX + C3 * (1.+2.*ADX)) * FX + C4 * CC
  FXX = FXX / RT
  Y(I) = Y(I) - PKHT * FXX
10 CONTINUE
RETURN
END

```

\$large

INTEGER*2 FUNCTION STRLEN(STRING)

```

C*****
C
C          STRLEN
C          =====
C*****
C
C      Developed for : Safeguards Technology Program
C                    Nuclear Chemistry Division
C                    Lawrence Livermore National Laboratory
C                    Livermore, CA. 94550
C
C      Developed by : COMPUchem, Inc.
C                   3099B Huntwood Avenue, Suite 107
C                   Hayward, CA. 94544
C                   (415)489-6514
C
C      This work supported by LLNL PO# 9107605
C*****
C
C      Title       : BWF
C      Revision    : 1.0
C      Author      : Carol Chen and Russell Fleming
C      Date        : 11/25/87
C      Language    : Microsoft Fortran Ver. 4.01
C*****
C
C      Abstract   : Returns actual string length of passed string
C*****
C
C      Arguments  : String
C*****
C
C      Routines   :
C      Called    : None
C
C      Libraries  :
C      Used      : None
C*****
C
C      Revision History ( include date, revision I.D., and details )
C*****
C
C      COMMON and variable declarations
C*****
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C*****
C
C      variable initialization declarations
C*****
C
C      character string*(*)
C      integer*2 index,length

```

```

1234      index=len(string)
        continue
        if( (string(index:index).eq.' ').and.(index.gt.1) )then
            index=index-1
            goto 1234
        endif
        STRLEN=index
        return
        end

```

\$large

subroutine testbin(filename,type,ndev)

TESTBIN

Developed for : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
 3099B Huntwood Avenue, Suite 107
 Hayward, CA. 94544
 (415)489-6514

This work supported by LLNL POW 9107605

Title : Testbin
 Revision : 1.0
 Author : Russell Fleming
 Date : 11/9/87
 Language : Microsoft Fortran Ver. 4.01

Abstract : Tests data file for binary or ASCII format.

Arguments : filename,type,ndev

Routines
 Called : None
 Libraries
 Used : None

Revision History (Include date,revision I.D., and details) *

COMMON and variable declarations *

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

variable initialization declarations *

C ___ Tests a data file for binary or ascii format

character type*1,filename*30,junk*8
 integer ndev*2

open(ndev,file=filename,status='OLD',iostat=istat)

20 read(ndev,20)junk
 format(a8)
 close(ndev)

type=' '
 do 10 i=1,8
 itemp=ichar(junk(i:i))
 if (((itemp.lt.48).or.(itemp.gt.57)).and.(itemp.ne.45)).and.
 + ((itemp.ne.46)) then
 type='B'
 end if
 10 continue
 if (type.ne.'B')type='A'

100 return
 write(*,*)' Error in opening file ',filename,' Press enter'
 pause ' '
 close(ndev)
 stop
 return
 end

---->

Slarge

SUBROUTINE TTEST(WPCT,RAT,NGRP,T2)

TTEST

=====

Developed by : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

This work supported by LLNL PO# 9107605

Title : TTEST
Revision : 2.0
Author : Wayne D. Ruhter
Revised for the IBM PC by Russell Fleming
COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :
This subroutine performs the statistical T*2 test to evaluate
the determined Pu-isotopic ratios against those declared by
the facility operator.

Arguments : WPCT,RAT,NGRP,T2

Routines
Called : None

Libraries
Used : None

Revision History (Include date, revision I.D., and details)

COMMON and variable declarations

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

variable initialization declarations

COMMON /MAIN/ X(9),

1 Y(210),
1 ALPHA(6,6),
1 SUM(6),
1 ERROR(7),
1 YFIT(10),
1 SIGMAA(7)

2 DIMENSION WPCT(6),RAT(4),BMAT(4,4),DFACT(4),
ORAT(4),DEL(4),PDEL(4)

C Corrections for half-lives and mass differences

C DATA DFACT/.003628,.27252,.0006009,.018025/

C IF (NGRP.EQ.4)NGRP=5

C T*2 TEST CANNOT BE PERFORMED, IF OPERATOR VALUES NOT GIVEN

C IF (WPCT(1) .LT. 0)THEN

T2 = -1

RETURN

C END IF

C IF (WPCT(2) .GT. 0.)GO TO 5

T2 = -1

C RETURN

C Transfer matrix

C DO 30 J = 1,4

C DO 30 K = 1,4

C BMAT(J,K) = ALPHA(J,K)

C CALL MATINV(BMAT,4,4,DET)

C Calculate operator ratios

C ORAT(1) = WPCT(1)/(WPCT(2)*DFACT(1))

C ORAT(2) = WPCT(3)/(WPCT(2)*DFACT(2))

C ORAT(3) = WPCT(4)/(WPCT(2)*DFACT(3))

C ORAT(4) = WPCT(6)/(WPCT(2)*DFACT(4))

C Calculate T*2

C NR = 4

C IF (ORAT(4) .EQ. 0)NR = 3

C T2 = 0.

C DO 50 J = 1,NR

C PDEL(J) = 0.

C DEL(J) = dLOG(ORAT(J)/RAT(J))

C DO 60 K = 1,NR

C DO 60 J = 1,NR

C PDEL(K) = PDEL(K) + DEL(J) * BMAT(J,K)

C DO 70 J = 1,NR

C T2 = T2 + PDEL(J) * DEL(J)

C RETURN

C END

---->

\$large

SUBROUTINE WRTEMP(tsnd,tpke,tarea,terr,tnpks,tndev,iln,prnt)

WrTemp
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL PGN# 9107605

Title : WrTemp
Revision : 1.0
Author : Wayne D. Ruhter
: Revised for the IBM PC by Russell Fleming
Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :
This subroutine will print out a line of output.

Arguments : tsnd,tpke,tarea,terr,tnpks,tndev

Routines
Called : None
Libraries
Used : None

Revision History (Include date,revision I.O., and details) *

COMMON and variable declarations *

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION TAREA(44),TERR(44)

variable initialization declarations *

CHARACTER *24 TSND

INTEGER*2 Tnpks,Tndev

C
C format declarations
C *****

333 FORMAT(F6.2,1X,F10.0,F6.2)
938 FOKMAT(1X,A)

WRITE(tsnd,333)TPKE,TAREA(TNPKS),TERR(TNPKS)
IF(TERR(TNPKS).EQ.0) TSND(19:)= ' FIXD '
WRITE(tndev,938)TSND
call more(iln,prnt)

RETURN
END

```
COMMON /INPUT/ nref,tp238,tp239,tp240,tp241,tp242,ta241,
x               tpe238,tpe239,tpe240,tpe241,tpe242,tae241,
x               coltim,fname,mame,bin,printer,exflag,rep,
x               spectime,specdate,amdate,pudate,sample

CHARACTER BIN*1,PRINTER*1,EXFLAG*1,RNAME*30,FNAME*30,JUNK*1
character rep*1,specdate*10,spectime*10,amdate*10,pudate*10
character sample*10

INTEGER*4      Nref(2)
```

common /top/ drivepath

character drivepath*45

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL PD# 9107605

Title : IAEA Main routine
Revision : 4.2
Author : Russel Fleming
Date : 9/26/88
Language : Clipper Summer '87

Abstract : Main control program for data acquisition, analysis,
and quality control. This program will run under
dBASE III+ and call compiled fortran programs.

Arguments :

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

Last Revision : 2/12/88

Interpreter Directives

```
set date ansi
set status off
set safety off
set talk off
set bell off
set device to screen
set print off
```

```
set scoreboard off
set deleted on
set cursor off
*Public flags
public firstflag, abort, flag

*Public variables
public drivepath, f1, f9, f10, finput, up, down, esc, cr, pgup, pgdn, bs
public val, delete, strg, r, c, maxcnt, lib, win, cl
```

```
set color to w/
```

variable initialization declarations

```
*--- Get Default drive and path
if !file('defaults.dbf')
  clear
  @ 2,0 say 'ERROR -- The file DEFAULTS.DBF Must be in the current directo
  ---->try'
  ?
  ? 'Unable to continue program until DEFAULTS.DBF is found. Correct and re
  ---->try.'
  ?
  ?
  if file('mast.bat')
    erase mast.bat
  endif
  if file('status.ia')
    erase status.ia
  endif
  if file('onetime.txt')
    erase onetime.txt
  endif
  return
endif
use defaults
go top
drivepath=trim(trim(drive))+trim(path))
use

if .not. file('onetime.txt')
  clear
  text
```

IAEA Plutonium Isotopics Analysis

by

The Lawrence Livermore National Laboratories

and

Compuchem, Inc.

Version 4.2

July 1989

Written by:

W.D. Ruhter R. Fleming R. Gunnink W.M. Buckley R. Wenning

endtext

```

@ 2,1 to 4,79 double
@ 6,14 to 12,65
@ 23,1 say ''

```

junk=inkey(5)

else

*---[Check IAEA.DBF for duplicate last record]---

```

use tempinfo
zap
append from switch.prm sdf
go top
if (stuff1='')
do checkia
endif
endif

```

endif

*---[If Onetime.txt exists, then have been here this run]---

*---[delete onetime.txt when exiting]---

```

set alte to onetime.txt
set alte on
set console off
?? ' '
set console on
close alternote

```

```

del=7
esc=27
cr=13
pgup=18
pgdn=3
up=5
down=24
bs=8
f9=-8
f10=-9
f1=28
f2=-1
f3=-2
f4=-3
f5=-4
f6=-5
f7=-6
f8=-7

```

```

finput=84
firstflag=f.
abort=f.
cup=chr(24)
cdown=chr(25)
selected=f.
menuch=1

```

```

*
*
* Main Loop

```

do while .t.

finput=0

clear

```

@ 0,0 SAY "IAEA Plutonium Analysis System"
@ 1,1 SAY date()
@ 1,72 SAY time()
@ 3,22 SAY "International Atomic Energy Agency"
@ 5,29 SAY "Plutonium Analysis"
@ 8,11 SAY "Menu Options:"

```

text

1) Transfer Data from Cicero Analyzer

A) Analyze Data from Gamma-Ray Spectra on disk

C) Check Quality Assurance

D) Default Settings

Q) Quit this menu and return to system

Enter Choice:

endtext

```

@ 2,8 TO 6,70
@ 6,8 TO 22,70
@ 6,8 say chr(195)
@ 6,70 say chr(180)
@ 24,7 say "<Enter the appropriate key or use the arrow keys>"
@ 24,55 say " then press enter>"
@ 20,39 say '!'

```

* Get function key response

menuch = 1

char = ""

set cursor off

do while .not. selected

do funkey

if finput=f1

do mainhelp with 'N',char

endif

do selmenu with menuch,selected,char

@ 20,39 SAY char

enddo

set cursor on

do case

case menuch=1

do c1cdata

if .not. abort

if file('switch.prm')

erase switch.prm

endif

if file('mast.bat')

erase mast.bat

endif

set alte to switch.prm

set alte on

set console off

??'C'

?drivepath

close alte

set alte to mast.bat

set alte on

?? 'Echo off'

```

? 'REM batch file for Transfer, returns to IAEA program'
? 'master'
? 'ia.bat'
?
close alte
set console on
clear
return
endif
abort=.f.
finput=0
selected = .f.
loop
case menuch=2
if file('status.ia')
firstflag=.t.
else
firstflag=.f.
endif
do getiadat
if .not. abort
if file('switch.prm')
erase switch.prm
endif
if file('mast.bat')
erase mast.bat
endif
set alte to switch.prm
set alte on
set console off
??'!'
?drivepath
close alte
set alte to mast.bat
set alte on
?? 'Echo off'
? 'REM batch file for Analysis, returns to IAEA program'
? 'master'
? 'ia.bat'
?
close alte
set console on
clear
return
endif
abort=.f.
finput=0
selected = .f.
loop
case menuch=3
do dch
if .not. abort
if file('switch.prm')
erase switch.prm
endif
if file('mast.bat')
erase mast.bat
endif
set alte to switch.prm
set alte on
set console off
??'!'
?drivepath
close alte
set alte to mast.bat
set alte on

```

```

?? 'Echo off'
? 'REM batch file for Dcheck, returns to IAEA program'
? 'master'
? 'ia.bat'
?
close alte
set console on
clear
return
endif
abort=.f.
finput=0
selected = .f.
loop
case menuch=4
* ... change default settings
do chdefalt
finput=0
use defaults
go top
drivepath=trim(trim(drive))+trim(path))
use
selected = .f.
loop
case menuch=5
clear
exit
endcase
select ed = .f.
enddo
set cursor on
if file('mast.bat')
erase mast.bat
endif
if file('status.ia')
erase status.ia
endif
if file('onetime.txt')
erase onetime.txt
endif
if file('switch.prm')
erase switch.prm
endif
do enditall
*
* --- EOF IAEA.PRG
*
procedure selmenu
parameters menuch,selected,char
if (finput=cr).or. (finput=pgdn)
selected = .t.
return
else
do case
case finput=84
char = "I"
menuch = 1
case finput=65
char = "A"
menuch = 2

```

```

        case finput=67
          char = "C"
          menuch = 3

        case finput=68
          char = "D"
          menuch = 4

        case finput=81
          menuch=5
          char = "Q"

        case finput = 5
          menuch=menuch-1
          if menuch<1
            menuch=5
          endif

        case finput = 24
          menuch=menuch+1
          if menuch>5
            menuch=1
          endif

        case finput = esc
          menuch=5
          selected =.t.

      endcase
    endif
    if finput = 5 .or. finput = 24
      do case
        case menuch = 1
          char ="T"
        case menuch = 2
          char ="A"
        case menuch = 3
          char ="C"
        case menuch = 4
          char ="D"
        case menuch = 5
          char ="Q"
      endcase
    endif
    return
*****
Procedure MainHelp
para lib,win
do help
return

```

```

*
*
* International Atomic Energy Agency
*
*
*
*
* Developed for : Safeguards Technology Program
*                 Nuclear Chemistry Division
*                 Lawrence Livermore National Laboratory
*                 Livermore, CA. 94550
*
* Developed by : COMPUchem, Inc.
*                 30998 Huntwood Avenue, Suite 107
*                 Hayward, CA. 94544
*                 (415)489-6514
*
* This work supported by LLNL POW 9107605
*
*
* Title : Getiadat
* Revision : 1.0
* Author : Russel Flemming and Richard Wenning
* Date : 9/26/88
* Language : Clipper Summer '87
*
*
* Abstract : Front end for analysis section
*
*
* Arguments :
*
*
* Routines
* Called : HELP,GETNUM,GETSTR,GETDATE,GETTIME
*
* Libraries
* Used :
*
*
*
* Revision History ( Include date,revision I.D., and details )
*
*
*

```

```
clear
```

```
use defaults
drivepath=trim(trim(drive)+trim(path))
use
set format to
```

```
* ___ Read last Cicero spectra file name from FILENAME.CIC
```

```
if .not. file('filename.cic')
```

```
  xspec=space(30)
```

```
else
```

```
  filenamedbf=drivepath+'filename'
```

```

if file(''+filenamedbf+'.dbf')
  use &filenamedbf
  zap
  filenamedbf=drivepath+'filename'
  use &filenamedbf
  append from filename.cic sdf
  nnn=filename
  do justname with nnn
  xspec=nnn
else
  @ 22,0 clear
  @ 22,0 say 'Error opening FILENAME.DBF. Check default drive and path set
---->tings'
  wait
  abort=.t.
  return
endif
endif
use

* --- first pass thru peak position form?
firstpas = .t.

*---[ NEEDREP is flag if need to derive report name from spectra file name ]--
*needrep=.t.

*---[ ABORT is flag for either exiting from IAEA and calling Fortran or not ]-
abort=.f.

*---[ FIRST is flag if need to get peak positions from NCAPAR ]---
*first=.f.

*---[ HOCKFN is flag for checking if need to derive report name from spectra ]
*hockfn=.f.

*---[ iaeainuse tells wether the iaea.dbf is active or not ]---
iaeainuse = .f.

*---[ BADFILE is the flag for whether an analysis file exists before analysis ]
badfile = .f.

*---[ SAVESCR is flag set to save screen values from analysis screen if pgdn is
---->hit]
savescr=.f.
* The existence of status.ia file determines the value of FIRSTFLAG.
* This value is given in iaea.prg and chdefalt.prg.
if .not. firstflag
  iadebdf=drivepath+'iadebdf'
  if file(''+iadebdf+'.dbf')
    use &iadebdf
    go top
    xout=output
  else
    @ 22,0 clear
    @ 22,0 say 'Error opening IADEFALT.DBF. Check default drive and path set
---->tings'
    wait
    abort=.t.
    return
  endif
else
  iaeabdf=drivepath+'iaea'
  if file(''+iaeabdf+'.dbf')
    use &iaeabdf
    go bottom
    iaeainuse = .t.

```



```

else
  @ 22,0 clear
  @ 22,0 say 'Error opening IAEA.DBF. Check default drive and path setting
....>s'
  wait
  abort=.t.
  return
endif
endif

* --- Set up first time through flag and write file
* --- if status.ia exists, we have been through here before and set flag

firstflag=.t.
set alte to status.ia
set alte on
set console off
?? ' '
set console on
close alternate

ctemp=' '
rep=''

* ___ First screen defaults
* ___ (if first pass, values obtained from iadefalt)
* ___ [otherwise from iaea.dbf]

xsample=space(10)
npos148=0
npos208=0
xrep=repfname
xout=output
xincl='N'

if iaeasinuse
  xmeasdate=measdate
  xmeasime=measime
else
  xmeasdate=ctod(' . . ')
  xmeasime=' : : '
endif

* ___ Second screen defaults
temp38=pu238
temp39=pu239
temp40=pu240
temp41=pu241
temp42=pu242
tempam=am241

temp38e=pu238err
temp39e=pu239err
temp40e=pu240err
temp41e=pu241err
temp42e=pu242err
tempame=am241err
xcooling=cooling

xpudate=pudate
xamdate=amdate

* ___ Third screen defaults
xbatch=batch
xcm=cmat

```

```

xcm2=cmat2
xcm3=cmat3
xno=noitem
xtpumas=tpumas
xmethod=method
xtumas=tumas
xenrich=enrich
xmdc=mdc

iaeadb=drivepath+'iae'
if file('&iaedb'+'.dbf')
  use &iaedb
else
  @ 22,0 clear
  @ 22,0 say 'Error opening IAEA.DBF. Check default drive and path settings'
  wait
  abort=.t.
  return
endif

append blank

* get the current record number so we know where to go if we have to jump
* through IAEA.dbf to find peak positions when not in mcapar.dbf.
tempcno = recno()

* ___ Replace with default values
replace pos148 with space(10)
replace pos208 with space(10)
replace specname with xspec
replace repfname with xrep
replace output with xout
replace sampleum with xsample
replace measdate with xmeasdate
replace measime with xmeasime

replace pu238 with temp38
replace pu239 with temp39
replace pu240 with temp40
replace pu241 with temp41
replace pu242 with temp42
replace am241 with tempam

replace pu238err with temp38e
replace pu239err with temp39e
replace pu240err with temp40e
replace pu241err with temp41e
replace pu242err with temp42e
replace am241err with tempame
replace cooling with xcooling
replace pudate with xpudate
replace amdate with xamdate

replace batch with xbatch
replace cmat with xcm
replace cmat2 with xcm2
replace cmat3 with xcm3
replace noitem with xno
replace tpumas with xtpumas
replace method with xmethod
replace tumas with xtumas
replace enrich with xenrich
replace mdc with xmdc

commit

```

```

scnum = 1  &&---[ Scnum is the screen number ]...
editing = .t. &&---[ Editing is flag for menu control loop ]...
firstin = .t. &&---[ Firstin enables going to first screen first, then menu ]...
flg=0      &&---[ Flg returns info from input routine ]...
fnum=1     &&---[ Fnum is the field number ]...

```

```

do while editing
  set color to w/
  if .not. firstin
    clear
    @ 0,1 say "IAEA Plutonium Analysis System"
    @ 1,1 say date()
    @ 1,72 say time()
    text

```

IAEA Plutonium Analysis

- F) File and Peak Information
- P) Plutonium Abundances
- B) Batch Sample Information
- A) Analyze data
- Q) Quit and Return to Main Menu

Enter Choice:

```

endtext
@ 4,24 to 6,64
@ 6,24 to 10,64
@ 6,24 say chr(195)
@ 6,64 say chr(180)
@ 24,5 say "Enter the appropriate key or use the arrow keys, "
@ 24,56 say "then press enter>"
@ 17,45 say "f"

```

```

* Get the menu choice
selected = .f.
menuch = 1
char = "f"
do while .not. selected
  do funky
    if finput=f1
      do panahelp with 'P',char
    endif
    do selAnal with menuch,selected,char
    @ 17,45 say char
  enddo

```

```

do case
  case menuch = 1
    scnum=1
    flg=0
  case menuch = 2
    scnum=2
  case menuch = 3
    scnum=3
  case menuch = 5
    scnum=0

```

```

if .not. savescr
  delete
endif
pack
abort=.t.
firstflg=.f.
return

```

```

case menuch = 4
  badfile=.f.
  flg=0
  do chkfile with specprm,flg,badfile
  if .not. badfile
    scnum = 0
    editing = .f.
  else
    scnum=1
  endif
endcase

```

```

ctemp=' '
endif
firstin=.f.

```

```

* -----
do case
  case scnum = 1
    set color to w/
    clear
    @ 0, 1 say "IAEA Plutonium Analysis System"
    @ 1, 1 say date()
    @ 1, 72 say time()
    @ 5, 25 SAY "IAEA Plutonium Analysis"
    @ 8, 20 SAY "Spectral File Name:"
    @ 10, 22 SAY "148 Kev Position:"
    @ 12, 22 SAY "208 Kev Position:"
    @ 14, 15 SAY "Output Report File Name:"
    @ 16, 10 SAY "Direct Report Output (S,P,D):"
    @ 16, 43 SAY "(Screen,Printer,Disk)"
    @ 18, 10 say "Include Peak Areas in Report (Y,N):"
    @ 4, 4 TO 6, 70
    @ 6, 4 TO 20, 70
    @ 6, 4 say chr(195)
    @ 6, 70 say chr(180)
    set color to /w
    if firstpas
      @ 8, 40 say space(30)
      @ 10, 40 say space(10)
      @ 12, 40 say space(10)
      @ 14, 40 say space(30)
      @ 16, 40 say space(1)
      @ 18, 46 say space(1)

```

```

mcapardbf=drivepath+mcapar'
if file('mcapardbf'+'.dbf')

```

```

  use mcapardbf
  locate for upper(fname)=upper(xspec) && was tempspec not xspec
  go bottom
  r=recno()
  do while (.not.(upper(fname)=upper(xspec) ) .and. ;
    (.not.bof() )
    skip -1

```

```

        enddo
        r=recno()
        go r
        if .not. bof()
            npos148=pos148
            npos208=pos208
            xmeasdate= measdate
            xmeasdate= measdate
            xsample = samplenum
        else
            xspec=space(30)
            npos148=0
            npos208=0
            xmeasdate=ctod(' . . . ')
            xmeasdate=' . . . '
        endif
        check for 's' at the beginning of file name
        do ckfname with xspec,xrep
        use &iaeadbf
        go temprecno
    endif
    set color to /w
    @ 8,40 say xspec
    @ 10,40 say npos148 picture '####.###'
    @ 12,40 say npos208 picture '####.###'
    @ 14,40 say xrep
    @ 16,40 say xout
    @ 18,46 say xincl
    set color to w/
    do inputban

    do while scnum=1
        set format to
        do case
            case fnum=1
                badfile=.t.
                do while (badfile .and. flg<=esc)
                    tspec=xspec
                    do isagetstr with xspec,8,40,30,flg,'P','C'
                *---[ If no drive or path given, use default drivepath ]---
                    ps=at('\',xspec)
                    pc=at(':',xspec)
                    if ( (ps=0) .and. (pc=0) )
                        chspec=trim(drivepath)+ltrim(trim(xspec))
                    else
                        chspec=ltrim(xspec)
                    endif
                    if tspec<>xspec
                *---[Check for file existence ]---
                        do chkfile with chspec,flg,badfile
                        firstpas = .t.
                    else
                *---[Check for file existence ]---
                        do chkfile with chspec,flg,badfile
                        firstpas = .f.
                    endif
                enddo
            *
            *---[ write variable for out_ia.prm ]---
            specprm=chspec

            set color to /w
            do case
                case flg=asc
                    xspec=iaea->specfname

```

```

        exit
    case flg=up
        if firstpas && tspec<>xspec
            first=.f.
            do justname with xspec
            do ckfname with xspec,xrep
            @ 14,40 say xrep
        endif
        do iodecf with fnum
    case (flg=cr .or. flg=down .or. flg=30 .or. flg=pgdn;
        .or. flg=pgup)
        if firstpas
            first=.f.
            do justname with xspec
            do ckfname with xspec,xrep
            @ 14,40 say xrep
        endif
        fnum=fnum+1
        if fnum=6
            fnum=1
        endif
    endcase
    set color to w/

    * ASSUMPTION: if can't find fname in mcapar database, blank out positions.
    * ----- can make it use last positions entered in analysis.
        if firstpas && .not. first
            mcapardbf=drivepath+'mcapar'
            if file('mcapardbf'+'.dbf')
                use mcapardbf
            else
                @ 22,0 clear
                @ 22,0 say "Error opening MCAPAR.DBF. Check default
    ----> drive and path settings!
                wait
                abort=.t.
                return
            endif
            go top
            fntemp=trim(xspec)
            do justname with fntemp
            locate for upper(trim(fntemp))=upper(trim(mcapar->fname))
    ---->)
        if .not. eof()
            npos148=pos148
            npos208=pos208
            xsample=samplenum
            xmeasdate=measdate
            xmeasdate=measdate
            set color to /w
            @ 10,40 say npos148 picture '####.###'
            @ 12,40 say npos208 picture '####.###'
            set color to w/
        else
            *---[If the sample number and 148,208 positions can't be found in ]---
            *---[MCAPAR.DBF, then search in IAEA.DBF file ]---
            *---[The values should be in IAEA if the data has been analyzed beforehand]---
            iaeadb = drivepath + 'iaea'
            use &iaeadbf
            go bottom
        *---[Loop through iaea until last occurrence of specfname ]---
            found = .f.
            recnum=0

```

```

do while .not. bof()
  if upper(trim(fntemp))=upper(trim(specfname))
    recnum=recno()
    exit
  endif
  skip -1
enddo

*---[if at beginning of the file then no record found]----
if .not. bof()

  go recnum
  tpos148 = pos148
  tpos208 = pos208
  xsample = samplenum
  xmeasdate = measdate
  xmeasdate = measdate

*---[strip characters from the field of the record and return a number
do justnum with tpos148, npos148
do justnum with tpos208, npos208

  set color to /w
  @ 10, 40 say npos148 picture '#####.###'
  @ 12, 40 say npos208 picture '#####.###'
  set color to /w

*---[Get the samplenum, date, time for the values not found in the iaec.dbf
*---[but are found in the mcapar.dbf
do smdttm with xsample, xmeasdate, xmeasdate, flg

  if flg=esc
    exit
  endif

*---[If the file name could not be found in the mcapar.dbf or the iaec.dbf,
*---[we query for the measurement date and time as well as the sample number.
else

  npos148 = 0
  npos208 = 0
  xsample = space(10)
  xmeasdate = ctime(' . . ')
  xmeasdate = ' : : '
  set color to /w
  @ 10, 40 say npos148 picture '#####.###'
  @ 12, 40 say npos208 picture '#####.###'

  set color to w/
  @ 23,0 clear
  @ 23,0 say trim(fntemp) + " Peak information is no

---->t found."

  wait

*---[Get the samplenum, date, time for the values not found in the iaec.dbf
*---[but are found in the mcapar.dbf
do smdttm with xsample, xmeasdate, xmeasdate, flg
if flg=esc
  exit
endif

endif

set color to w/
@ 21,0 clear
@ 22,0 clear
@ 23,0 clear

```

```

  @ 24,0 clear
  do inputban

endif

iaecdbf=drivepath+'iaec'
use &iaecdbf
go tempreno
first = .t.

*---[Endif for (.not. first)
endif
if flg=pgdn
  exit
endif
case frum=2
  p148=npos148
  set color to /w
  @ 10, 40 say npos148 picture '#####.###'
  @ 12, 40 say npos208 picture '#####.###'
  set color to w/
  do iagetnum with p148, 10, 40, 8, flg, 3, .f., 'P', 'C'
do case
case flg=esc
  exit
case flg=up
  npos148=p148
  frum=frum-1
  if frum=1
    frum=6
  endif
case (flg=cr .or. flg=down .or. flg=10)
  npos148=p148
  frum=frum-1
  if frum=1
    frum=6
  endif
case (flg=pgun .or. flg=pgup)
  npos148=p148
  exit
endcase
case frum=3
  frum=frum-1
  set color to /w
  @ 10, 40 say npos148 picture '#####.###'
  @ 12, 40 say npos208 picture '#####.###'
  set color to w/
  do iagetnum with p208, 12, 40, 8, flg, 3, .f., 'P', 'C'
do case
case flg=esc
  exit
case flg=up
  npos208=p208
  do iadecf with frum
case (flg=cr .or. flg=down .or. flg=10)
  npos208=p208
  frum=frum-1
  if frum=6
    frum=1
  endif
case (flg=pgdn .or. flg=pgup)

```

```

npos208=p208
exit
endcase
case fnum=4
do iagetstr with xrep,14,40,30,flg,'P','C'
*---[ If no drive or path given, use default drivepath ]---
* ps=at('\',rep)
* pc=at(':',rep)
* if ( (ps=0) .and. (pc=0) )
* spec=trim(drivepath)+trim(rep)
* endif
do case
case flg=esc
xrep=iaea->repfname
exit
case flg=up
xrep=rep
do iadecf with fnum
case (flg=cr .or. flg=down.or. flg=30)
xrep=rep
fnum=fnum+1
if fnum>6
fnum=1
endif
case (flg=pgdn .or. flg=pgup)
xrep=rep
exit
endcase
case fnum=5
beenthere=.f.
errorflag=.f.
do while at(xout,'SPD')=0 .or. .not. beenthere
if errorflag
do inputba
errorflag=.f.
endif
beenthere=.t.
do iaentr1 with xout,16,40,flg
do while flg=f1
do panahelp with 'P','C'
do iaentr1 with xout,16,40,flg
enddo
xout=upper(xout)
do case
case flg=esc
xout=iaea->output
exit
case at(xout,'SPD')=0
if flg <> esc
xout=' '
@ 22,0 clear
@ 22,0 say 'Invalid Output Device, Enter an S,
errorflag=.t.
wait
loop
endif
---->P or D'

```

75

```

case flg=up
do iadecf with fnum
case (flg=cr .or. flg=down)
fnum=fnum+1
if fnum>6
fnum=1
endif
case (flg=pgdn .or. flg=pgup)
exit
otherwise
if at(xout,'SPD')=0
xout=' '
@ 22,1 clear
@ 22,1 say 'Invalid Output Device, Enter an S,
---->P or D'
wait
errorflag=.t.
loop
endif
fnum=fnum+1
if fnum>6
fnum=1
endif
endcase
enddo
if flg=esc .or. flg=pgdn
exit
endif
case fnum=6
set color to /w
@ 18, 46 say xincl
set color to /w
do iaentr1 with xincl,18,46,flg
do while flg=f1
do panahelp with 'P','C'
do iaentr1 with xincl,18,46,flg
enddo
xincl=upper(xincl)
do case
case flg=esc
xincl='M'
exit
case flg=up
do iadecf with fnum
case at(xincl,'M')=0
xincl='M'
case (flg=cr .or. flg=down)
fnum=fnum+1
if fnum>6
fnum=1
endif
case (flg=pgdn .or. flg=pgup)
exit
endcase
enddo
if at(xout,'SPDspd')=0
?? chr(7)
@ 22,1 clear
@ 22,1 say 'Invalid Output Device, Enter an S,P or D'
errorflag=.t.
wait

```

```

do inputban
loop
endif
trm=chspec
do justname with trm
if flgpgdn .or. flgpgup
go tempreno
replace specname with trm
replace pos148 with str(npos148,8,3)
replace pos208 with str(npos208,8,3)
replace repfname with xrep
replace output with xout
replace samplenum with xsample

replace pu238 with temp38
replace pu239 with temp39
replace pu240 with temp40
replace pu241 with temp41
replace pu242 with temp42
replace am241 with tempam

replace pu238err with temp38e
replace pu239err with temp39e
replace pu240err with temp40e
replace pu241err with temp41e
replace pu242err with temp42e
replace am241err with tempame
replace xcooling with xcooling
replace xupdate with xupdate
replace xamdate with xamdate

replace batch with xbatch
replace xmat with xcm
replace xmat2 with xcm2
replace xmat3 with xcm3
replace xnoitem with xno
replace xtpumas with xtpumas
replace xmethod with xmethod
replace xtumas with xtumas
replace xenrich with xenrich
replace xmdc with xmdc
replace xmeasdate with xmeasdate
replace xmeas time with xmeas time

commit
save scr=.t.
endif

```

```

.....
case scnum=2
ctemp=' '
set color to w/
@ 0, 1 say "IAEA Plutonium Analysis System"
@ 1, 1 say date()
@ 1, 72 say time()
@ 4, 4 to 6, 68
@ 6, 4 to 21, 68
@ 6, 4 say chr(195)
@ 6, 68 say chr(180)
@ 9, 22 say "Enter Plutonium Abundances"

@ 7, 6 say "Pu Analysis Date:"
@ 7, 37 say "Am Analysis Date:"

```

```

@ 9, 6 say "%Pu-238 Abundance:"
@ 9, 37 say "%Pu-238 % Error:"
@ 11, 6 say "%Pu-239 Abundance:"
@ 11, 37 say "%Pu-239 % Error:"
@ 13, 6 say "%Pu-240 Abundance:"
@ 13, 37 say "%Pu-240 % Error:"
@ 15, 6 say "%Pu-241 Abundance:"
@ 15, 37 say "%Pu-241 % Error:"
@ 17, 6 say "%Pu-242 Abundance:"
@ 17, 37 say "%Pu-242 % Error:"
@ 19, 6 say "%Am-241 Abundance:"
@ 19, 37 say "%Am-241 % Error:"
do inputban

temp=0
set color to w/
@ 0, 1 say "IAEA Plutonium Analysis System"
@ 1, 1 say date()
@ 1, 72 say time()
@ 22, 1 clear
coolf=' '
@ 23, 1 say "Are the abundances known (Y,N) ?"
set color to w/
@ 23, 34 say coolf pict 'X'
junk=0
do while at(chr(junk),'Y|N')=0 .and. junk<>esc .and. junk<>cr
junk=inkey(1)
if junk=ft
do panahelp with 'P','E'
endif
endif
enddo
if junk=cr
coolf='Y'
endif
set color to w/
coolf=chr(junk)
if .not. lastkey()=esc
coolf=upper(coolf)
if coolf = 'N'
clexcooling
do getcool
xcooling=c1
temp38=0
temp39=0
temp40=0
temp41=0
temp42=0
tempam=0
temp38e=0
temp39e=0
temp40e=0
temp41e=0
temp42e=0
tempame=0
lkey=0
else
set color to w/
@ 7, 25 say xUPDATE
@ 7, 55 say xAMDATE
@ 9, 25 say temp38 PICTURE "####.####"
@ 11, 25 say temp39 PICTURE "####.####"
@ 13, 25 say temp40 PICTURE "####.####"
@ 15, 25 say temp41 PICTURE "####.####"
@ 17, 25 say temp42 PICTURE "####.####"
@ 19, 25 say tempam PICTURE "####.####"

```

```

a 9, 53 SAY temp38E PICTURE "###.####"
a 11, 53 SAY temp39E PICTURE "###.####"
a 13, 53 SAY temp40E PICTURE "###.####"
a 15, 53 SAY temp41E PICTURE "###.####"
a 17, 53 SAY temp42E PICTURE "###.####"
a 19, 53 SAY tempame PICTURE "###.####"
do inputban

FIELDNUM=1
FLG=0
DONE = .f.
DO WHILE .NOT. DONE
  DO CASE
    CASE FIELDNUM=1
      do !agetdate with xupdate,7,25,flg,'P','E'

    CASE FIELDNUM=2
      do !agetdate with xamdate,7,55,flg,'P','E'

    CASE FIELDNUM=3
      DO !Agetnum WITH temp38,9,25,10,flg,4,.t.,'P','E'

    CASE FIELDNUM=4
      DO !Agetnum WITH temp39,11,25,10,flg,4,.t.,'P','E'

    CASE FIELDNUM=5
      DO !Agetnum WITH temp40,13,25,10,flg,4,.t.,'P','E'

    case fieldnum=6
      DO !Agetnum WITH temp41,15,25,10,flg,4,.t.,'P','E'

    CASE FIELDNUM=7
      DO !Agetnum WITH temp42,17,25,10,flg,4,.t.,'P','E'

    CASE FIELDNUM=8
      DO !Agetnum WITH tempam,19,25,10,flg,4,.t.,'P','E'

    CASE FIELDNUM=9
      DO !Agetnum WITH temp38e,9,53,8,flg,4,.t.,'P','E'

    CASE FIELDNUM=10
      DO !Agetnum WITH temp39e,11,53,8,flg,4,.t.,'P','E'

    CASE FIELDNUM=11
      DO !Agetnum WITH temp40e,13,53,8,flg,4,.t.,'P','E'

    case fieldnum=12
      DO !Agetnum WITH temp41e,15,53,8,flg,4,.t.,'P','E'

    CASE FIELDNUM=13
      DO !Agetnum WITH temp42e,17,53,8,flg,4,.t.,'P','E'

    CASE FIELDNUM=14
      DO !Agetnum WITH tempame,19,53,8,flg,4,.t.,'P','E'

  ENDCASE
DO CASE
  CASE FLG=UP
    FIELDNUM=FIELDNUM+1
    IF FIELDNUM<1
      FIELDNUM=14
    ENDIF
  CASE FLG=DOWN .OR. FLG=CR
    FIELDNUM=FIELDNUM+1
    IF FIELDNUM>14
      FIELDNUM=1
  ENDIF

```

```

        ENDIF
        CASE FLG=ESC
          EXIT
        case FLG=pgdn
          exit
        otherwise
          fieldnum=fieldnum+1
          if fieldnum>14
            fieldnum=1
          endif
        ENDCASE
      ENDDO
    endif
  endif
* -----
*   Third screen
*   Case scnum=3
*   ctemp=1
*   set color to w/
*   clear
a 0, 1 say "IAEA Plutonium Analysis System"
a 1, 1 say date()
a 1, 72 say time()
a 4, 28 say "Batch Sample Information"
a 3, 3 to 5, 75
a 5, 3 to 12, 75
a 5, 3 say chr(195)
a 5, 75 say chr(180)
a 12, 3 to 19, 75
a 12, 3 say chr(195)
a 12, 75 say chr(180)
a 6, 7 SAY "Batch Name:"
a 6, 39 SAY "Canning Materials:"
a 8, 7 SAY "Number of Items:"
a 10, 7 SAY "Material Description:"
a 13, 7 SAY "Plutonium Mass:"
a 13, 44 SAY "Uranium Mass:"
a 15, 15 SAY "Plutonium Analysis Method:"
a 17, 20 SAY "Uranium Enrichment:"

do inputban

a 6, 19 SAY xBATCH
a 6, 58 SAY xCM
a 7, 58 SAY xCM2
a 8, 58 SAY xCM3
a 8, 24 SAY xMO
a 10, 29 SAY xMOC
a 13, 23 SAY xTPUMAS
a 13, 58 SAY xTPUMAS
a 15, 42 SAY xMETHOD
a 17, 40 SAY xENRICH
set color
fnum=1
notdone=.t.
do while notdone
  numgood=.f.
  do case
    case fnum=1
      do !agetstr with xbatch,6,19,15,flg,'P','G'

    case fnum=2
      do !agetstr with xcm,6,58,15,flg,'P','G'

    case fnum=3
      do !agetstr with xcm2,7,58,15,flg,'P','G'

```

```

      case fnum=4
        do iagetstr with xcm3,8,58,15,flg,'P','G'
      case fnum=5
        do iagetstr with xno,8,24,5,flg,'P','G'
      case fnum=6
        do iagetstr with xmdc,10,29,15,flg,'P','G'
      case fnum=7
        do iagetnum with xtpumas,13,23,12,flg,3,.t.,'P','G'
      case fnum=8
        do iagetnum with xtumas,13,58,12,flg,3,.t.,'P','G'
      case fnum=9
        do iagetstr with xmethod,15,42,20,flg,'P','G'
      case fnum=10
        do iagetstr with xenrich,17,40,10,flg,'P','G'
    endcase
  do case
    case flg=up
      fnum=fnum-1
      if fnum<1
        fnum=10
      endif
    case flg=down .or. flg=cr
      fnum=fnum+1
      if fnum>10
        fnum=1
      endif
    case flg=esc
      xBATCH = iaea->batch
      xCM = iaea->cmat
      xCM2 = iaea->cmat2
      xCM3 = iaea->cmat3
      xNO = iaea->noitem
      xMDC = iaea->mdc
      xTPUMAS = iaea->tpumas
      xTUMAS = iaea->tumas
      xMETHOD = iaea->method
      xENRICH = iaea->enrich
      exit
    case flg=pgdn
      exit
    endcase
  enddo
endcase
enddo

*---[ Replace current values ]-----

xpos148=str(npas148,8,3)
xpos208=str(npas208,8,3)
replace pos148 with xpos148
replace pos208 with xpos208
replace measdate with xmeasdate
replace measstime with xmeasstime

xx=chspec
do justname with xx
replace specname with xx

```

```

replace repfname with xrep
replace output with xout

replace pu238 with temp38
replace pu239 with temp39
replace pu240 with temp40
replace pu241 with temp41
replace pu242 with temp42
replace am241 with tempam
replace pu238err with temp38e
replace pu239err with temp39e
replace pu240err with temp40e
replace pu241err with temp41e
replace pu242err with temp42e
replace am241err with tempame
replace cooling with xcooling
replace samplerum with xsample

replace batch with xbatch,cmat with xcm,cmat2 with xcm2,;
      cmat3 with xcm3,noitem with xno,tpumas with xtpumas,;
      pudate with xpudate,amdate with xamdate,method with xmethod
replace tumas with xtumas,enrich with xenrich,mdc with xmdc

commit
*---[ Now write this data to OUT_IA.PRM ]-----

if file('out_ia.prm')
  erase out_ia.prm
endif

set alternate to out_ia.prm
set alternate on
set console off

ntemp=0
do justnum with iaea->pos148,ntemp
stemp=space(2)+str(ntemp,6)
? stemp

*num=int(log(val(iae->pos148))/log(10))+1
*stemp=space(10-num)+substr(iae->pos148,1,num)

do justnum with iaea->pos208,ntemp
stemp=space(2)+str(ntemp,6)
? stemp

*num=int(log(val(iae->pos208))/log(10))+1
*stemp=space(10-num)+substr(iae->pos208,1,num)

? iaea->pu238
? iaea->pu239
? iaea->pu240
? iaea->pu241
? iaea->pu242
? iaea->am241
? iaea->pu238err
? iaea->pu239err
? iaea->pu240err
? iaea->pu241err
? iaea->pu242err
? iaea->am241err
? iaea->cooling
if atc(' ',str(cooling))=0

```



```

?? '0'
endif
? specprm
? iaea->repfname
? iaea->output
? xinc1
? xmeasdate
? xmeasime
? xpuddate
? xandate
? xsample
?

close alternate
set console on
close databases
clear
use
return

*
* -- End of routine GETIADA.PRG
*

*****
*--- Test for first character of file name for an 'S'
*
* If found, report name is the data file name with an 'R' on front
*****
procedure ckfname
parameters xsp,rep

indx=1
maxcnt=len(xsp)
sp=trim(xsp)
stemp=substr(sp,1,1)

* --- strip off leading blanks
do while stemp=' '
  indx = indx+1
  sp = substr(sp,indx,len(sp)-1)
  stemp = substr(sp,1,1)
enddo

if upper(substr(sp,1,1))='S'
  rep='R'+substr(sp,2,len(sp)-1)
  rep=rep+space(maxcnt-len(rep))
else
  rep=''
endif

return

*****
* Decrement Fnum in Getiadat
*****
procedure iadecf
parameters fn

fn=fn-1
if fn<=0
  fn=6
endif
return

```

```

*****
* --- input routine for one character field
*
*****
procedure iaeinrl
parameters strg,r,c,flag

set cursor on

oldstrg=strg
set color to /w
@ r,c say ''

ch=inkey(.1)

do while (ch<>esc .and. ch<>or .and. ch<>pgup .and. ch<>pgdn .and.;
  if (ch>31)
    strg=chr(ch)
    @ r,c say strg
    @ r,c say ''
  else
    if ch=f1
      flag=f1
      set cursor off
      return
    endif
    ch=inkey(.1)
  enddo
  flag=ch
  set color to w/
  if ch=esc
    strg=oldstrg
  endif
  set cursor off
  return

*****
*---Chkfile
*---This procedure checks for the existance of the file.
*---if the file does not exist it warns the user of that fact.
*****
procedure chkfile
parameters spec,flg,badfile

if .not. file(spec) .and. flg <> esc
  @ 22,0 clear
  @ 22,0 say "File not found, please check spelling or the default drive path"
  wait
  @ 22,0 clear
  @ 23,0 clear
  do inputban
    badfile = .t.
  else
    badfile = .f.
  endif
  return

*****
*---Justname
*---This procedure strips the pathname,if one exists, from the
*---spectrum file name

```

```

*****
procedure justname
parameters stg

start=1
nums=at('\',stg)
numc=at(':',stg)

if numc>nums .or. (numc=0.and.numc>0) && Colon is after slash or no slash
  stg=substr(stg,numc+1,len(stg)-numc)
else
  if nums>0
    do while (nums>0).or.(numc>0)
      if numc>0
        stg=substr(stg,numc+1,len(stg)-numc)
      endif
      nums=at('\',stg)
      numc=at(':',stg)
    enddo
  endif
endif
return

```

```

*****
*---SelAnal
*---This procedure selects the menu choice
*****
procedure selAnal
para menuch,selected,char

```

```

If (finput=er) .or. (finput=PgDn)
  selected = .t.
  return
else
  do case
    case finput = 70
      char = "F"
      menuch = 1
    case finput = 80
      char = "P"
      menuch = 2
    case finput = 66
      char = "B"
      menuch = 3
    case finput = 65
      char = "A"
      menuch = 4
    case finput = 81
      char = "Q"
      menuch = 5
    case finput = 5
      menuch=menuch-1
      if menuch<1
        menuch = 5
      endif
    case finput = 24
      menuch=menuch+1
      if menuch>5
        menuch = 1
      endif
  endcase

```

```

endif
case finput = esc
  menuch=5
  selected = .t.
endcase
endif
if finput = 5 .or. finput = 24
  do case
    case menuch = 1
      char = "F"
    case menuch = 2
      char = "P"
    case menuch = 3
      char = "B"
    case menuch = 4
      char = "A"
    case menuch = 5
      char = "Q"
    endcase
  endif
return

```

```

*****
procedure panahelp
para lib,win
do help
return

```

```

*****
procedure lagetnum
para val,r,c,maxcnt,flag,dec,prerr,lib,win
numgood=.f.
do while numgood=.f.
  do getnum
    if val<0
      numgood=.f.
    else
      numgood=.t.
    endif
  enddo
set color to w/
if prerr
  if .not. numgood
    @ r+1,c say 'Re-enter'
  else
    @ r+1,c say ' '
  endif
endif
enddo
return

```

```

*****
procedure lagetdate
para date,r,c,flag,lib,win
do getdate
return

```

```

*****
procedure lagetime
para strg,r,c,flag,lib,win
do gettime
return

```

```

*****

```

```

procedure iagetstr
para str,r,c,maxcnt,flag,lib,win
do getstrg
do while flag=f1
  do help
  do getstrg
enddo
return

```

```
*****
```

```

procedure justnum
para xinput,noutput

```

```

ntemp=''
lp=1
do while lp<len(xinput)
  ch=substr(xinput,lp,1)
  if at(ch,'1234567890.')<>0
    ntemp=ntemp+ch
  endif
  lp=lp+1
enddo
noutput=val(ntemp)
return

```

```

*****
*SMOITH
* This procedure gets the samplerum(sm),date(dt),and time(tm) for the
* iaea database file if the program can not find that data in the mcarp
* database file.
*****

```

```

procedure smdttm
para str,dt,time,flg

```

```

set color to w/
@ 22,0 clear to 24,79
@ 22,5 say "Enter sample number:"
@ 23,5 say "Date of Measurement:"
@ 24,5 say "Time of Measurement:"
set color to /w
@ 22,26 say str picture 'XXXXXXXXXX'
@ 23,26 say dt
@ 24,26 say time
set color to w/

```

```

fld=1
flg=0

```

```

do while .t.
  do case
    case fld=1
      do iagetstr with str,22,26,10,flg,'P','C'
    case fld=2
      do iagetdate with dt,23,26,flg,'P','C'
    case fld=3
      do iagetime with time,24,26,flg,'P','C'
  endcase
  do case
    case flg=esc
      exit
    case flg=up
      fld=fld-1

```

```

    flg=0
    if fld=0
      fld=1
    endif
    otherwise
      if fld=3
        exit
      endif
      fld=fld+1
    endcase
  enddo
return

```

```

*****
*_____ End Of File GETIADAT.PRQ ._.
*****

```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL PG# 9107605

Title : Decnum
Revision : 1.0
Author : Russell Fleming
Date : 11/9/87
Language : Clipper summer '87

Abstract : Decrements field pointer for Cicdata front end

Arguments :

Routines
Called :
Libraries
Used :

Revision History (include date, revision I.D., and details)

```
fnun=fnun-1
if fnun<0
  fnun=17
endif
return
---->
```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30996 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : Inputban
Revision : 1.0
Author : Russell Fleming
Date : 9/26/88
Language : Clipper Summer '87

Abstract : Prints a banner at the bottom for Read inputs

Arguments :

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

set color to w/

```
@ 23,1 say ' Type In Value          '+
chr(24)+'', '+chr(25)+' and '+chr(17)+chr(196)+chr(217)
@ 24,1 say '                          Move Field Cursor'
@ 23,50 say ' PgDn          Esc'
@ 24,50 say 'Exit Screen  Abort Screen'
set color to /w
```

return

*
* --- End of file INPUTBAN.PRG

78

International Atomic Energy Agency

Developed for : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
 30998 Huntwood Avenue, Suite 107
 Hayward, CA. 94544
 (415)489-6514

This work supported by LLNL PC# 9107605

Title : Getcool
 Revision : 1.0
 Author : Russell Fleming
 Date : 9/26/88
 Language : Clipper Summer '87

Abstract : Gets the cooling time for Getladat front end

Arguments :

Routines
 Called :
 Libraries
 Used :

Revision History (Include date, revision I.D., and details)

```
lib='P'
win='E'
cl=4
cool=.t.
```

```
@ 21, 7 SAY "Cooling Time (Years):"
@ 22, 13 SAY "(if known)"
@ 21, 29 GET cl PICTURE "#####"
read
```

```
return
```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POM 9107605

Title : Enditall
Revision : 1.0
Author : Russell Fleming
Date : 9/26/88
Language : Clipper Summer '87

abstract : Ends system run. Prints ending message

Arguments :

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

set format to
close data
20,1
text

Ending IAEA Plutonium Acquisition and Analysis Program.

endtext
?

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCKEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL PQ# 9107605

Title : Getkey
Revision : 1.0
Author : Russell Fleming
Date : 9/26/88
Language : clipper Summer '87

Abstract : Gets a key from the keyboard ignoring all other
key strokes.

This routine only returns:
CR, ESC, Alt-F10, F9, *10, Up/Down arrows, f1, f2, f3, pgdn
13 27 -39 -8 -9 5 24

Arguments :

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

```

finput=0
do while .not. (finput= cr .or.;
                finput= esc .or.;
                finput= f1 .or.;
                finput= f2 .or.;
                finput= f3 .or.;
                finput= f9 .or.;
                finput= f10 .or.;
                finput= pgdn .or.;
                finput= up .or.;
                finput= down .or.;
                finput= pgdn .or.;
                finput= inkey(.1)
            )

```

```

finput= f10 .or.;
finput= up .or.;
finput= down .or.;
finput= pgdn)

```

finput=inkey(.1)

enddo
return

* End of file GETKEY.PRG

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCKEN, Inc.
3099B Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL PO# 9107605

Title : Dch
Revision : 1.0
Author : Russell Fleming
Date : 9/26/88
Language : Clipper Summer '87

Abstract : Front End for Dcheck quality assurance

Arguments :

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

```
dcheckdbf=drivepath+'dcheck'
if file('&dcheckdbf'+'.dbf')
    use &dcheckdbf
else
    @ 22,0 clear
    @ 22,0 say 'Error opening DCHECK.DBF.   Check default drive and path settings
    ...>'
    wait
    abort=.t.
    return
endif

go bottom
tfname=name
```

```
tnref1=tnref1
tnref2=tnref2
tstd1=tstd1
tstd2=tstd2
```

```
set format to
clear
@ 0, 0 SAY "IAEA Plutonium Analysis System"
@ 1, 1 SAY date()
@ 1, 72 SAY time()
@ 5, 25 SAY "Quality Assurance Check"
@ 7, 11 SAY "Low Energy Peak Position:"
@ 8, 24 SAY "Peak Energy:"
@ 13, 10 SAY "High Energy Peak Position:"
@ 14, 24 SAY "Peak Energy:"
@ 18, 7 SAY "Binary Data File Name:"

set color to w/
@ 7, 37 SAY tnref1
@ 8, 37 SAY tstd1
@ 13, 37 SAY tnref2
@ 14, 37 SAY tstd2
@ 18, 30 SAY tfname

set color to w/
@ 4, 6 TO 6, 76
@ 6, 6 TO 19, 76
@ 6, 6 say chr(195)
@ 6, 76 say chr(180)

do inputban
set color to w/
```

```
***[ Process Binary File Name for \ or : ]***
```

```
fnum=1
done=.f.
flg=0
do while .not. done
do case
case fnum=1
do dgetwnum with tnref1, 7,37,11,flg,'D','X'
case fnum=2
do dgetnum with tstd1, 8,37,11,5,flg,'D','X'
case fnum=3
do dgetwnum with tnref2,13,37,11,flg,'D','X'
case fnum=4
do dgetnum with tstd2,14,37,11,5,flg,'D','X'
case fnum=5
do dgetstr with tfname,18,30,30,flg,'D','X'
endcase
do case
case flg=esc
abort=.t.
set color to w/
return
case flg=down .or. flg=sr
fnum=(fnum+1)
if fnum=5
fnum=1
endif
case flg=up
fnum=fnum-1
if fnum=1
fnum=5
endif
case flg=pgdn
ps=at('\',tfname)
pc=at(':',tfname)
```

```

*---[ Didn't find a : or \, use default drivepath ]---
      tfname=trim(drivepath)+trim(tfname)
    endif
    replace fname with tfname
    replace nref1 with tnref1
    replace nref2 with tnref2
    replace std1 with val(str(tstd1,11,5))
    replace std2 with val(str(tstd2,11,5))

    * write values to temporary file for dcheck.for
    if file('dcheck.prm')
      erase dcheck.prm
    endif

    set alternate to dcheck.prm
    set alte on
    set console off

    if tnref1>0
      num=int((log(tnref1)/log(10))+1)
      stemp=space(10-num)+str(tnref1,num)
    else
      stemp=' 0.0'
    endif
    ?? stemp

    if tnref2>0
      num=int((log(tnref2)/log(10))+1)
      stemp=space(10-num)+str(tnref2,num)
    else
      stemp=' 0.0'
    endif
    ? stemp

    ? tstd1
    ? tstd2
    ? tfname

    close alternate
    set console on

    use
    set color to w/
    exit
  endcase
enddo
return

```

```

*****
procedure dgetwnum
para val,r,c,maxcnt,flag,lib,win
do getwnum
return

```

```

procedure dgetnum
para val,r,c,maxcnt,dec,flag,lib,win
do getnum
return

```

```

procedure dgetstr
para strg,r,c,maxcnt,flag,lib,win
do getstrg
do while flag=f1

```

```

do help
do getstrg
enddo
return

```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POM 9107605

Title : CHECKIA
Revision : 1.0
Author : Russell Fleming
Date : 1/12/88
Language : Clipper Autumn 1986

Abstract : Checks last record of ANRESULT.DBF and if that info
exists before, the first record is replaced with the
last record.

Arguments : None

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

debug=.f.

```
anres=drivepath+'anresult'
if file('&anres+'.dbf)
    use &anres
else
    @ 22,0 clear
    @ 22,0 say 'Error opening ANRESULT.DBF. Check default drive and path setting
...->s'
    wait
    abort=.t.
    return
```

endif

go bottom
recno=recno()

samp=trim(upper(sample))
filen=trim(upper(specfname))

x1=specfname
x2=sample
x3=pu238
x4=pu239
x5=pu240
x6=pu241
x7=pu242
x8=pe238
x9=pe239
x10=pe240
x11=pe241
x12=pe242
x13=pcw
x14=pcwe
x15=e240
x16=e240e
x17=alpha11
x18=alpha12
x19=alpha13
x20=alpha14
x21=alpha22
x22=alpha23
x23=alpha24
x24=alpha33
x25=alpha34
x26=alpha44
x27=zx4849
x28=zx4849
x29=zx4049
x30=zx4049
x31=zx4149
x32=zx4149
x33=zx5141
x34=zx5141
x35=zx2549
x36=zx2549
x37=numcounts
x38=a241
x39=ae241
x40=mu235
x41=mu235
x42=fuwm

go top
locate for (trim(upper(sample))=samp) .and. (trim(upper(specfname))=filen)
.and. (recno()<>recno)

eof=eof()

```
if (debug)
    if (eof)
        ?'Didn't find it, must be a new record'
    else
        ?'Found record at record number "'
        ?? recno()
        ??', bottom recno='
        ?? recno
    endif
```

```

if ( (feof) .and. (recno<>recno()) )
  replace specfname with x1,sample with x2,pu238 with x3,pu239 with x4
  replace pu240 with x5,pu241 with x6,pu242 with x7,pe238 with x8
  replace pe239 with x9,pe240 with x10,pe241 with x11,pe242 with x12
  replace pow with x13,powe with x14,e240 with x15,e240e with x16
  replace alpha11 with x17,alpha12 with x18,alpha13 with x19,alpha14 with x20
  replace alpha22 with x21,alpha23 with x22,alpha24 with x23,alpha33 with x24
  replace alpha34 with x25,alpha44 with x26,z4849 with x27,e24849 with x28
  replace z4049 with x29,e24049 with x30,z4149 with x31,e24149 with x32
  replace z5141 with x33,e25141 with x34,z2549 with x35,e22549 with x36
  replace nuncounts with x37,a241 with x38,ee241 with x39,u235 with x40
  replace ue235 with x41,fwhm with x42
  go bottom
  delete
  pack

  if (debug)
    ?"Deleting last record"
    ??eof = "
    ??eof
    ??" recno = "
    ?? recno
    ??" Recno()="
    ??recno()
  endif

endif

use

if (debug)
  wait
endif

re!rn

*
**** End of file CHECKIA.PRG

```

```

*
* -----
*
* International Atomic Energy Agency
*
* -----
*
* Developed for : Safeguards Technology Program
*                 Nuclear Chemistry Division
*                 Lawrence Livermore National Laboratory
*                 Livermore, CA. 94550
*
* Developed by : COMPUCHEN, Inc.
*                 30998 Huntwood Avenue, Suite 107
*                 Hayward, CA. 94544
*                 (415)489-6514
*
* This work supported by LLNL POW 9107605
*
* -----
*
* Title       : CICDATA
* Revision    : 1.0
* Author      : Russel Fleming
* Date       : 9/26/88
* Language    : Clipper Summer '87
*
* -----
*
* Abstract   : Main control program for data acquisition, analysis,
*               and quality control. This program will run as an
*               executable file and call compiled fortran programs.
*
* -----
*
* Arguments :
*
* -----
*
* Routines
* Called   :
*
* Libraries
* Used     :
*
* -----
*
* Revision History ( Include date, revision I.D., and details )
*
* -----

```

```

* both flags publically defined and explained in isaa.prg
abort=.f.
flag=0

```

```
declare s[10]
```

```
set intensity on
set scoreboard off

```

```
mcapardbf=drivepath+'mcapar'
if file('&mcapardbf'+'.dbf')

```

```

use &mcapardbf
else
  a 22,0 clear
  a 22,0 say 'Error opening MCPAR.DBF. Check default drive and path settings
  ....>'
  wait
  abort=.t.
  return
endif

set format to
choice=0

* ___ Load defaults
go bottom
xpur=pur
xrate=rate
xpza=pza
xinpul=inpul
xblrl=blrl
xblrs=blrs
xds=dts
xinpmd=inpmd
xbase=base
xrtps=rtps
xhvpul=hvpul
xshapet=shapet
xcgain=cgain
xfgain=fgain
xrtptr=rtptr
xhvlt=hvlt
xdiscr=discr
xcntry=cntry
xfclty=fclty
xmba=mba
xkmp=kmp
xreport=report
xdate=date
insptr=insptr
xrmks=rmks
xsamplenum=samplenum
rpos208=pos208
rpos148=pos148
xfname=fname
xmeasdate=date()
xmeastime=time()

* ___ Make a place for the new data
append blank

* ___ Put defaults into new record
replace pur with xpur,blrl with xblrl,blrs with xblrs,rate with xrate,,
hvlt with xhvlt,hvpul with xhvpul,discr with xdiscr,,
pza with xpza,inpul with xinpul,shapet with xshapet
replace dts with xds,inpmd with xinpmd,base with xbase,rtptr with xrtptr,,
rtps with xrtps,cgain with xcgain,fgain with xfgain,,
cntry with xcntry,fclty with xfclty,mba with xmba,kmp with xkmp
replace report with xreport,date with xdate,insptr with xinsptr,,
rmks with xrmks
replace samplenum with xsamplenum,pos208 with rpos208,pos148 with rpos148,,
fname with xfname,measdate with xmeasdate,meastime with xmeastime

* ___ Decide current indexes for each field for first screen
commit

if upper(trim(xpur))='OFF'

```

File: C1CDATA.PRG

07/25/1989 08:37 -- Page: 003

```

else
  ind1=1
endif
if upper(trim(xblt1))='THR'
  ind2=2
else
  ind2=1
endif
if upper(trim(xblrs))='ASYM'
  ind3=2
else
  ind3=1
endif
if upper(trim(xrate))='HIGH'
  ind4=2
else
  ind4=1
endif
if upper(trim(xhvpol))='MEG'
  ind5=2
else
  ind5=1
endif
if upper(trim(xinpol))='AMP-'
  ind9=3
else
  if upper(trim(xinpol))='ADC'
    ind9=2
  else
    ind9=1
  endif
endif
endif
do case
  case xshapet=10.0
    ind10=6
  case xshapet=6.0
    ind10=5
  case xshapet=3.0
    ind10=4
  case xshapet=2.0
    ind10=3
  case xshapet=1.0
    ind10=2
  otherwise
    ind10=1
endcase
if upper(trim(xdts))='FIX'
  ind11=2
else
  ind11=1
endif
if upper(trim(xinpmid))='SAMP'
  ind12=2
else
  ind12=1
endif
do case

```

File: C1CDATA.PRG

07/25/1989 08:37 -- Page: 004

```

  case xrtpt=25.0
    ind14=7
  case xrtpt=16.0
    ind14=6
  case xrtpt=8.0
    ind14=5
  case xrtpt=6.0
    ind14=4
  case xrtpt=3.0
    ind14=3
  case xrtpt=1.5
    ind14=2
  otherwise
    ind14=1
endcase
if upper(trim(xrtps))='PEAK'
  ind15=2
else
  ind15=1
endif
do case
  case cgain=2000
    ind16=9
  case cgain=1000
    ind16=8
  case cgain=500
    ind16=7
  case cgain=200
    ind16=6
  case cgain=100
    ind16=5
  case cgain=50
    ind16=4
  case cgain=20
    ind16=3
  case cgain=10
    ind16=2
  otherwise
    ind16=1
endcase
* ___ Main Loop for all screens
needdata = .t.
scnum = 3
do while needdata
  frum=1
  if scnum=1
    set color to w/
    clear
    @ 0,1 say "IAEA Plutonium Analysis System"
    @ 1,1 say date()
    @ 1,72 say time()
    @ 3,19 say "IAEA Plutonium Acquisition System Setup"
    @ 2, 0 to 4, 79
    @ 6, 24 say "Adj.:"
    @ 6, 47 say "High Voltage : "
    @ 7, 1 say "PUR: "
    @ 7, 18 say "BLR Shape: "
    @ 7, 47 say "Polarity : "
    @ 8, 18 say "... Rate: "
    @ 10, 39 say "Low-Level Discriminator: "
    @ 11, 8 say "PZA Adjustment: "
    @ 13, 8 say "Input Switch: "

```

```

@ 13, 52 SAY "..... ADC ....."
@ 15, 8 SAY "Shaping Time:"
@ 14, 47 SAY "Dead Time ::"
@ 15, 47 SAY "Input Mode::"
@ 18, 13 SAY "..... GAIN ....."
@ 17, 47 SAY "Base Line Comment::"
@ 19, 9 SAY "Coarse ::"
@ 21, 9 SAY "Fine Adjustment::"
@ 20, 47 SAY "RTP (micro sec)::"
@ 21, 57 SAY "RTPS ::"
@ 6,0 TO 8,12
@ 5,16 TO 9,36
@ 5,45 TO 8,76
@ 9,38 TO 11,78
@ 10,5 TO 16,34
@ 17,5 TO 22,41
@ 12,44 TO 22,78
set color to /w

* Display default values
@ 7, 7 say xpur
@ 6, 31 say xblrl
@ 7, 31 say xblrs
@ 8, 31 say xrate
@ 6, 62 say xhvlt
@ 7, 62 say xhvpot
@ 10, 64 say xdiscr
@ 12, 10 say xprz
@ 13, 22 say xirpot
@ 15, 22 say xshapet
@ 14, 59 say xdtc
@ 15, 59 say xirpmd
@ 18, 52 say xbase
@ 20, 64 say xrtpt
@ 21, 64 say xrtps
@ 19, 26 say xcgain
@ 21, 26 say xfgain
endif

do while scnum=1
set format to
do case
case fnum=1
* set up array with choice
s[1]='ON '
s[2]='OFF'

do pur with choice
do case
* F2, Next choice
case choice=1
ind1=ind1+1
if ind1>2
ind1=1
endif
xpur=s[ind1]

* Down arrow or Enter, next field
case (choice=24 .or. choice=13)
xpur=s[ind1]
fnum=fnum+1
if fnum>17
fnum=1
endif

* Up Arrow, previous field
case choice=5
xpur=s[ind1]

```

```

do decfnum
* PgDn, Accept all values and exit screen
case choice=PgDn
xpur=s[ind1]
exit

* Escape, abort out of screen, (save no results)
case choice=27
xpur=mcapar->pur
exit
case choice=f1
do cichelp with 'C','B'
loop
endcase
@ 7, 7 say xpur

* .....

case fnum=2
s[1]='AUTO'
s[2]='THR '

do blrd; with choice
do case
* F2 Next choice
case choice=1
ind2=ind2+1
if ind2>2
ind2=1
endif
xblrl=s[ind2]

* Down arrow or enter
case (choice=13 .or. choice=24)
xblrl=s[ind2]
fnum=fnum+1
if fnum>17
fnum=1
endif

* Up arrow
case choice=5
xblrl=s[ind2]
do decfnum
case choice=PgDn
xblrl=s[ind2]
exit

* Escape, don't save results
case choice=27
xblrl=mcapar->blrl
exit
case choice=f1
do cichelp with 'C','B'
loop
endcase
@ 6,31 say xblrl

* .....

case fnum=3
s[1]='SYN '
s[2]='ASYN'

do blrs with choice
do case
* F2 Next choice
case choice=1
ind3=ind3+1
if ind3>2
ind3=1

```

```

endif
xblrs=s[ind3]
* Down arrow or enter
case (choice=13 .or. choice=24)
xblrs=s[ind3]
fnum=fnum+1
if fnum>17
fnum=1
endif
* Up arrow
case choice=5
xblrs=s[ind3]
do decfnum
* PgDn, Accept all values
case choice=PgDn
xblrs=s[ind3]
exit
* Escape, don't save results
case choice=27
xblrs=mcapar->blrs
exit
case choice=f1
do c1chelp with 'C','B'
loop
endcase
a 7, 31 say xblrs
* .....
case fnum=4
s[1]='AUTO'
s[2]='HIGH'
do rate with choice
do case
* F2, Next choice
case choice=-1
ind4=ind4+1
if ind4>2
ind4=1
endif
xrate=s[ind4]
* Down arrow or enter
case (choice=13 .or. choice=24)
xrate=s[ind4]
fnum=fnum+1
if fnum>17
fnum=1
endif
* Up arrow
case choice=5
xrate=s[ind4]
do decfnum
* PgDn, Accept all values
case choice=PgDn
xrate=s[ind4]
exit
* Escape, don't save results
case choice=27
xrate=mcapar->xrate
exit
case choice=f1
do c1chelp with 'C','B'
loop
endcase
a 8, 31 say xrate
* --- HERE ---

```

```

case fnum=5
set color to w/
a 6, 47 SAY "High Voltage :"
do inputban
if type('xhvlt')='M'
st=str(xhvlt,12,2)
else
st=xhvlt
endif
do dumpdsp with st
a 6,62 say st+space(12-len(st)+1)
do entr with st,6,62,12,flag,'C','B'
do case
case flag=99
clear
? "error in entr, flag=99"
wait
return
case flag=Esc
xhvlt=mcapar->xhvlt
exit
case flag=up
xhvlt=st
do decfnum
case (flag=cr .or. flag=down .or. flag=12)
xhvlt=st
fnum=fnum+1
if fnum>17
fnum=1
endif
endif
case flag=pgdn
xhvlt=st
exit
otherwise
xhvlt=st
fnum=fnum+1
if fnum>17
fnum=1
endif
endif
endcase
if type('st')='M'
xhvlt=st
else
xhvlt=val(st)
endif
set color to w/
a 6, 47 SAY "High Voltage :"
set color to /w
* .....
case fnum=6
s[1]='POS'
s[2]='NEG'
do polarity with choice
do case
* F2 Next choice
case choice=-1
ind6=ind6+1
if ind6>2
ind6=1
endif
endif

```



```

xhvpol=s[ind6]
* Down arrow or enter
case (choice=13 .or. choice=24)
xhvpol=s[ind6]
fnum=fnum+1
if fnum>17
fnum=1
endif
* Up arrow
case choice=5
xhvpol=s[ind6]
do decfnum
* PgDn, Accept all values
case choice=PgDn
xhvpol=s[ind6]
exit
* Escape, don't save results
case choice=27
xhvpol=rcapar->hvpol
exit
case choice=ff
do cichelp with 'C','B'
loop
endcase
a 7,62 say xhvpol
* -----
* HERE
case fnum=7
* Enter discriminator number here
set color to w*/
a 10,39 SAY "Low-Level Discriminator:"
do inputban
st=str(discr,12,2)
do dumpdsp with st
a 10,64 say st+space(12-len(st)+1)
do entr with st,10,64,12,flag,'C','B'
do case
case flag=99
clear
?"error in entr, flag=99"
wait
return
case flag=Esc
xdisc=rcapar->discr
exit
case flag=up
xdisc=st
do decfnum
case (flag=cr .or. flag=down .or. flag=12)
xdisc=st
fnum=fnum+1
if fnum>17
fnum=1
endif
case flag=PgDn
xdisc=st
exit
otherwise
xdisc=st
fnum=fnum+1
if fnum>17

```

```

fnum=1
endif
endcase
if type('st')='W'
xdisc=st
else
xdisc=val(st)
endif
set color to w/
a 10,39 SAY "Low-Level Discriminator:"
set color to /w
* HERE
case fnum=8
* Enter PZA adjustment field here
set color to w*/
a 11,8 SAY "PZA Adjustment:"
do inputban
a 12,10 say trim(xpza)+space(20-len(trim(xpza))+1)
st:cpza
do entr with st,12,10,20,flag,'C','B'
do case
case flag=99
clear
?"error in entr, flag=99"
wait
return
case flag=Esc
xpza=rcapar->pza
exit
case flag=up
xpza=st
do decfnum
case (flag=cr .or. flag=down .or. flag=12)
xpza=st
fnum=fnum+1
if fnum>17
fnum=1
endif
case flag=PgDn
xpza=st
exit
otherwise
fnum=fnum+1
if fnum>17
fnum=1
endif
endcase
xpza=st
set color to w/
a 11,8 SAY "PZA Adjustment:"
set color to /w
* -----
case fnum=9
a[1]='AMP'
a[2]='ADC'
a[3]='AMP'
do input with choice

```

```

do case
* F2 Next choice
  case choice=1
    ind9=ind9+1
    if ind9>3
      ind9=1
    endif
    xinp0=s[ind9]
* Down arrow or enter
  case (choice=13 .or. choice=24)
    xinp0=s[ind9]
    frum=frum+1
    if frum>17
      frum=1
    endif
* Up arrow
  case choice=5
    xinp0=s[ind9]
    do decfrum
* PgDn, Accept all values
  case choice=PgDn
    xinp0=s[ind9]
    exit
* Escape, don't save results
  case choice=27
    xinp0=mcapar->xinp0
    exit
  case choice=f1
    do cichelp with 'C','B'
    loop
  endcase
  @ 13, 22 say xinp0
* .....
  case frum=10
    s[1]=0.5
    s[2]=1.0
    s[3]=2.0
    s[4]=3.0
    s[5]=4.0
    s[6]=10.0
  do shapet with choice
  do case
* F2, Next choice
  case choice=1
    ind10=ind10+1
    if ind10>6
      ind10=1
    endif
    xshapet=s[ind10]
* Down arrow or enter
  case (choice=13 .or. choice=24)
    xshapet=s[ind10]
    frum=frum+1
    if frum>17
      frum=1
    endif
* Up arrow
  case choice=5
    xshapet=s[ind10]
    do decfrum
* PgDn, Accept all values
  case choice=PgDn
    xshapet=s[ind10]
    exit

```

```

* Escape, don't save results
  case choice=27
    xshapet=mcapar->xshapet
    exit
  case choice=f1
    do cichelp with 'C','B'
    loop
  endcase
  @ 15, 22 say xSHAPET
* .....
  case frum=11
    s[1]="VAR"
    s[2]="FIX"
  do dts with choice
  do case
* F2 Next choice
  case choice=1
    ind11=ind11+1
    if ind11>2
      ind11=1
    endif
    xdt=s[ind11]
* Down arrow or enter
  case (choice=13 .or. choice=24)
    xdt=s[ind11]
    frum=frum+1
    if frum>17
      frum=1
    endif
* Up arrow
  case choice=5
    xdt=s[ind11]
    do decfrum
* PgDn, Accept all values
  case choice=PgDn
    xdt=s[ind11]
    exit
* Escape, don't save results
  case choice=27
    xdt=mcapar->xdt
    exit
  case choice=f1
    do cichelp with 'C','B'
    loop
  endcase
  @ 14, 59 say xdt
* .....
  case frum=12
    s[1]="DC"
    s[2]="SAW"
  do inpmd with choice
  do case
* F2 Next choice
  case choice=1
    ind12=ind12+1
    if ind12>2
      ind12=1
    endif
    xinpmd=s[ind12]
* Down arrow or enter
  case (choice=13 .or. choice=24)
    xinpmd=s[ind12]

```

```

        fnum=fnum+1
        if fnum>17
            fnum=1
        endif
* Up arrow
        case choice=5
            xinpmd=s[ind12]
            do decfnum
* PgDn, Accept all values
            case choice=PgDn
                xinpmd=s[ind12]
                exit
* Escape, don't save results
            case choice=27
                xinpmd=mcapar->inpmd
                exit
            case choice=f1
                do cichelp with 'C','B'
                loop
            endcase
            @ 15, 59 say xinpmd
* -----
* _____ HERE _____
        case fnum=13
            * _____ Baseline comment
            set color to w/
            @ 17, 47 SAY "Base Line Comment:"
            do inputbn
            @ 18,52 say trim(base)+space(20-len(trim(base))+1)
            st=base
            do entr with st,18,52,20,flag, 'C','B'
            do case
                case flag=99
                    clear
                    ?"error in entr, flag=99"
                    wait
                    return
                case flag=Esc
                    xbase=mcapar->base
                    exit
                case flag=up
                    xbase=st
                    do decfnum
                case (flag=scr .or. flag=down .or. flag=12)
                    xbase=st
                    fnum=fnum+1
                    if fnum>17
                        fnum=1
                    endif
                case flag=pgdn
                    xbase=st
                    exit
                otherwise
                    xbase=st
                    fnum=fnum+1
                    if fnum>17
                        fnum=1
                    endif
            endcase
            xbase=st
            set color to w/

```

```

            @ 17, 47 SAY "Base Line Comment:"
            set color to w/
* -----
        case fnum=14
            s[1]=0.5
            s[2]=1.5
            s[3]=3.0
            s[4]=6.0
            s[5]=8.0
            s[6]=16.0
            s[7]=25.0
            do rtp with choice
            do case
* F2 Next choice
                case choice=-1
                    ind14=ind14+1
                    if ind14>7
                        ind14=1
                    endif
                    xrtpt=s[ind14]
* Down arrow or enter
                case (choice=13 .or. choice=24)
                    case
                        xrtpt=s[ind14]
                        fnum=fnum+1
                        if fnum>17
                            fnum=1
                        endif
* Up arrow
                    case choice=5
                        xrtpt=s[ind14]
                        do decfnum
* PgDn, Accept all values
                    case choice=PgDn
                        xrtpt=s[ind14]
                        exit
* Escape, don't save results
                    case choice=27
                        xrtpt=mcapar->xrtpt
                        exit
                    case choice=f1
                        do cichelp with 'C','B'
                        loop
                    endcase
                    @ 20, 64 say xrtpt
* -----
        case fnum=15
            s[1]="RTP "
            s[2]="PEAK"
            do rtps with choice
            do case
* F2 Next choice
                case choice=-1
                    ind15=ind15+1
                    if ind15>2
                        ind15=1
                    endif
                    xrtps=s[ind15]
* Down arrow or enter
                case (choice=13 .or. choice=24)
                    case
                        xrtps=s[ind15]
                        fnum=fnum+1
                        if fnum>17

```

```

      frnum=1
    endif
* Up arrow
    case choice=5
      xrtps=s[ind15]
      do decfrnum
* PgDn, Accept all values
      case choice=PgDn
        xrtps=s[ind15]
        exit
* Escape, don't save results
      case choice=27
        xrtps=mcapar->rtps
        exit
      case choice=f1
        do cichelp with 'C','B'
        loop
      endcase
      @ 21, 64 say XRTPS
* -----
      case frnum=16
        s[1]=5
        s[2]=10
        s[3]=20
        s[4]=50
        s[5]=100
        s[6]=200
        s[7]=500
        s[8]=1000
        s[9]=2000
      do cgain with choice
      do case
* F2 Next choice
        case choice=1
          ind16=ind16+1
          if ind16>9
            ind16=1
          endif
          xcgain=s[ind16]
* Down arrow or enter
        case {choice=13 .or. choice=24}
          xcgain=s[ind16]
          frnum=frnum+1
          if frnum>17
            frnum=1
          endif
* Up arrow
        case choice=5
          xcgain=s[ind16]
          do decfrnum
* PgDn, Accept all values
        case choice=PgDn
          xcgain=s[ind16]
          exit
* Escape, don't save results
        case choice=27
          xcgain=mcapar->cgain
          exit
        case choice=f1
          do cichelp with 'C','B'
          loop
        endcase
        @ 19, 27 say xcgain

```

```

* ----- HERE -----
      case frnum=17
        set color to w/*
        @ 21, 9 SAY "Fine Adjustment:"
        do inputban
        st=str(fgain,12,2)
        do dumpdsp with st
        @ 21,26 say st+space(12-len(st)+1)
        do entr with st,21,26,12,flag,'C','B'
        do case
          case flag=-99
            clear
            ?"Error in entr, flag=-99"
            wait
            return
          case flag=Esc
            xfgain=mcapar->fgain
            exit
          case flag=up
            xfgain=st
            do decfrnum
          case {flag=cr .or. flag=down .or. flag=12}
            xfgain=st
            frnum=frnum+1
            if frnum>17
              frnum=1
            endif
          case flag=PgDn
            xfgain=st
            exit
          otherwise
            xfgain=st
            frnum=frnum+1
            if frnum>17
              frnum=1
            endif
          endcase
          xfgain=val(st)
          set color to w/
          @ 21, 9 SAY "Fine Adjustment:"
          set color to /w
        endcase
      enddo
* ----- Second Screen -----
      if scnum=2
        set color to w/
        clear
        @ 0, 1 say "IAEA Plutonium Analysis System"
        @ 1, 1 say date()
        @ 1, 72 say time()
        @ 6, 3 to 10, 77
        @ 10, 3 to 18, 77
        @ 4, 3 to 6, 77
        @ 10, 3 say chr(195)
        @ 10, 77 say chr(180)
        @ 6, 3 say chr(195)
        @ 6, 77 say chr(180)
        @ 5, 26 SAY "Inspection Identification"
        @ 7, 6 SAY "Country Code:"
        @ 7, 40 SAY "Facility Code:"

```

```

@ 9, 6 SAY "Material Balance Code:"
@ 9, 40 SAY "Key Measurement Point Code:"
@ 11, 5 SAY "Inspection Report Number:"
@ 11, 47 SAY "Date of Measurement:"
@ 12, 47 SAY "Time of Measurement:"
@ 14, 5 SAY "Name of Coordinating Inspector:"
@ 16, 5 SAY "Remarks:"
set color to w/
@ 7, 20 say xcentry picture "XXXXXXXX"
@ 7, 35 say xfclty
@ 9, 29 say xmba
@ 9, 68 say xkmp
@ 11, 31 say xreport
@ 11, 68 say xmeasdate
@ 12, 68 say xmeastime picture "XXXXXXXX"
@ 14, 37 say xinspr
@ 17, 7 say xmrks
set color to w/
do inputba
set color to w/
fnum=1
notdone=.t.
flg=0
do while notdone
  do case
    case fnum=1
      do entr with xcentry,7,20,8,flg,'C','C'

    case fnum=2
      do entr with xfclty,7,35,6,flg,'C','C'

    case fnum=3
      do entr with xmba,9,29,6,flg,'C','C'

    case fnum=4
      do entr with xkmp,9,68,6,flg,'C','C'

    case fnum=5
      do entr with xreport,11,31,12,flg,'C','C'

    case fnum=6
      do cicgetdate with xmeasdate,11,68,flg,'C','C'

    case fnum=7
      do cicgettime with xmeastime,12,68,flg,'C','C'

    case fnum=8
      do entr with xinspr,14,37,24,flg,'C','C'

    case fnum=9
      do entr with xmrks,17,7,69,flg,'C','C'
  endcase
do case
  case flg=up
    fnum=fnum-1
    if fnum<1
      fnum=9
    endif
  case flg=down .or. flg=c
    fnum=fnum+1
    if fnum>9
      fnum=1
    endif
  case flg=esc
    exit
  case flg=pgdn

```

```

      exit
    endcase
  enddo
  set format to
endif

* ____ Third Screen "Sample Information"
if xnum=3
  set color to w/
  clear
  fexists=.t.
  do while fexists
    set color to w/
    clear
    set color to w/
    @ 0,1 say "IAEA Plutonium Analysis System"
    @ 1,1 say date()
    @ 1,72 say time()
    @ 5, 27 SAY "Sample/Spectrum Information"
    @ 8, 23 SAY "Sample Number:"
    @ 10, 15 SAY "148 KeV Peak Position:"
    @ 12, 15 SAY "208 KeV Peak Position:"
    @ 14, 18 SAY "Spectrum File Name:"
    set color to w/
    @ 8, 38 say xsamplenum
    @ 10, 38 say npos148 picture "####.###"
    @ 12, 38 say npos208 picture "####.###"
    @ 14, 38 say xfname
    set color to w/
    @ 4, 6 TO 6, 70
    @ 6, 6 TO 16, 70
    @ 6,6 say chr(195)
    @ 6,70 say chr(150)
    do inputba
    fnum=1
    notdone=.t.
    flg=0
    do while notdone
      do case
        case fnum=1
          do entr with xsamplenum,8,38,10,flg,'C','D'

        case fnum=2
          do cicgetnum with npos148,10,38,8,flg,3,.f.,'C','D'

        case fnum=3
          do cicgetnum with npos208,12,38,8,flg,3,.f.,'C','D'

        case fnum=4
          do entr with xfname,14,38,30,flg,'C','D'
      endcase
    do case
      case flg=up
        fnum=fnum-1
        if fnum<1
          fnum=9
        endif
      case flg=down .or. flg=c
        fnum=fnum+1
        if fnum>9
          fnum=1
        endif
      case flg=esc
        exit
      case flg=pgdn

```

```

*               exit
*replace pur with xpur,blrl with xblrl,blrs with xblrs,rate with xrate,;
*   hvolt with xhvolt,hvpot with xhvpot,discr with xdiscr,;
*   pza with xpza,inpul with xinpul,shapet with xshapet
*replace dts with xdis,inspm with xinspm,base with xbase,rtpst with xrtpt,;
*   rtps with xrtps,cgain with xcgain,fgain with xfgain,;
*   cntry with xcnyr,fclty with xfclty,mbs with xmba,kmp with xkmp
*replace report with xreport,date with xdate,insptr with xinsptr,;
*   rmrks with xrmrks
*replace samplenum with xsamplenum,pos208 with npos208,pos148 with npos148,;
*   fname with xftemp,measdate with xmeasdate,measstime with xmeasstime
*commit

```

```

               endcase
            enddo
            set color to w/
            set format to
            if flg=esc
                exit
            endif
*---[ Check if file name has : or \, Use name entered if so ]---
            pc=at(':',xfname)
            ps=at('\',xfname)
            if (pc=0) .and. (ps=0)
*---[ Use default drivepath ]---
                xfname=trim(drivepath)+trim(xfname)
            endif
* --- test for existence of data file name
            if file(xfname)
                yn='N'
                set color to w/
                @ 22,0 clear
                @ 22,0 say trim(xfname)+' exists! Do you want to overwrite it ? (Y,
***>N) ' get yn pict '!'
                read
                yn=upper(yn)
                if (yn='Y' .or. yn='y')
                    fexists=.t.
                else
                    fexists=.f.
                endif
            else
                fexists=.f.
            endif
            endif
        enddo
    endif

* ___ Get which screen to display
    set color to w/
    clear
    do nextsc with menuch
    do case
        case menuch= 1
            scnum=1
        case menuch= 2
            scnum=2
        case menuch= 3
            scnum=3
        case menuch= 4
            abort=.f.
            xftemp=xfname
            do justname with xftemp
    replace pur with xpur,blrl with xblrl,blrs with xblrs,rate with xrate,;

```

```

        hvolt with xhvolt,hvpot with xhvpot,discr with xdiscr,;
        pza with xpza,inpul with xinpul,shapet with xshapet
    replace dts with xdis,inspm with xinspm,base with xbase,rtpst with xrtpt,;
        rtps with xrtps,cgain with xcgain,fgain with xfgain,;
        cntry with xcnyr,fclty with xfclty,mbs with xmba,kmp with xkmp
    replace report with xreport,date with xdate,insptr with xinsptr,;
        rmrks with xrmrks
    replace samplenum with xsamplenum,pos208 with npos208,pos148 with npos148,;
        fname with xftemp,measdate with xmeasdate,measstime with xmeasstime
commit

```

```

        set format to
* ___ Write Cicero.Prm containing file name for spectra data
        set alte to cicero.prm
        set alte on
        set console off
        ??xfname
        ?
        set console on
        close alte
        exit
        select 5
        filename=drivepath+'filename'
        if file(filename+'.dbf')
            use &filename
        else
            @ 22,0 clear
            @ 22,0 say 'Error opening FILENAME.DBF. Check default drive and
----> path settings!
                wait
                abort=.t.
                return
            endif
            go top
            replace filename with xfname
            commit
            use
            select 1
            case menuch=5
                abort=.t.
            exit
        endcase
        set color to /w
    enddo
    set color to w/
    return

* _____ End of Main Routine _____

*****
* C1CDATA menu for Clipper compiled code
*
* Last Update: 9/12/88
*****

*****
* PUR Menu
*****
procedure pur
parameter ch

set color to w/
@ 7, 1 SAY "PUR:"

set color to w/

```

```

@ 23,1 say '      F2                '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice          Move Field Cursor'
@ 23,50 say ' PgDn                Esc'
@ 24,50 say 'Exit Screen  Abort Screen'

do getkey
ch:=input

set color to w/
@ 7, 1 SAY "PUR:"

set color to /w
return

*****
*      BLRADJ Menu
*****
procedure blradj
parameter ch

ch:=1
set color to w*/
@ 6, 24 SAY "Adj:~"
set color to w/

@ 23,1 say '      F2                '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice          Move Field Cursor'
@ 23,50 say ' PgDn                Esc'
@ 24,50 say 'Exit Screen  Abort Screen'

do getkey
ch:=input

set color to w/
@ 6, 24 SAY "Adj:~"
set color to /w
return

*****
*      BLR Shape Menu
*****
procedure blrs
parameter ch

ch:=1

set color to w*/
@ 7, 24 SAY "Shape:"
set color to w/

@ 23,1 say '      F2                '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice          Move Field Cursor'
@ 23,50 say ' PgDn                Esc'
@ 24,50 say 'Exit Screen  Abort Screen'

do getkey
ch:=input

set color to w/
@ 7, 24 SAY "Shape:"
set color to /w

```

```

return

*****
*      BLR Rate Menu
*****
procedure rate
parameter ch
ch:=1

set color to w*/
@ 8, 24 SAY "Rate:"
set color to w/

@ 23,1 say '      F2                '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice          Move Field Cursor'
@ 23,50 say ' PgDn                Esc'
@ 24,50 say 'Exit Screen  Abort Screen'

do getkey
ch:=input

set color to w/
@ 8, 24 SAY "Rate:"
set color to /w
return

*****
*      Polarity Menu
*****
procedure polarity
parameter ch

set color to w*/
@ 7, 47 SAY "Polarity  ~"
set color to w/

@ 23,1 say '      F2                '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice          Move Field Cursor'
@ 23,50 say ' PgDn                Esc'
@ 24,50 say 'Exit Screen  Abort Screen'

do getkey
ch:=input

set color to w/
@ 7, 47 SAY "Polarity  ~"
set color to /w
return

*****
*      Input polarity Menu
*****
procedure inpol
parameter ch

set color to w*/
@ 13, 8 SAY "Input Switch:~"
set color to w/

@ 23,1 say '      F2                '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choices        Move Field Cursor'
@ 23,50 say ' PgDn                Esc'

```

```
@ 24,50 say 'Exit Screen  Abort Screen'
```

```
do getkey
ch:=input
```

```
set color to w/
@ 13, 8 SAY "Input Switch:"
set color to /w
return
```

```
*****
*      Shape Time Menu
*****
procedure shapet
parameter ch
```

```
set color to w/
@ 15, 8 SAY "Shaping Time:"
set color to w/
```

```
@ 23,1 say '          F2          '+chr(24)+' '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice      Move Field Cursor'
@ 23,50 say ' PgDn          Esc'
@ 24,50 say 'Exit Screen  Abort Screen'
```

```
do getkey
ch:=input
```

```
set color to w/
@ 15, 8 SAY "Shaping Time:"
set color to /w
return
```

```
*****
*      DTS Menu
*****
procedure dts
parameter ch
```

```
set color to w/
@ 14, 47 SAY "Dead Time :"
```

```
set color to w/
@ 23,1 say '          F2          '+chr(24)+' '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice      Move Field Cursor'
@ 23,50 say ' PgDn          Esc'
@ 24,50 say 'Exit Screen  Abort Screen'
```

```
do getkey
ch:=input
```

```
set color to w/
@ 14, 47 SAY "Dead Time :"
```

```
set color to /w
return
```

```
*****
*      Inpmd Menu
*****
procedure inpmd
parameter ch
```

```
set color to w/
```

```
@ 15, 47 SAY "Input Mode:"
set color to w/
```

```
@ 23,1 say '          F2          '+chr(24)+' '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice      Move Field Cursor'
@ 23,50 say ' PgDn          Esc'
@ 24,50 say 'Exit Screen  Abort Screen'
```

```
do getkey
ch:=input
```

```
set color to w/
@ 15, 47 SAY "Input Mode:"
set color to /w
return
```

```
*****
*      RTP Menu
*****
procedure RTP
parameter ch
```

```
set color to w/
@ 20, 47 SAY "RTP (micro sec):"
```

```
set color to w/
@ 23,1 say '          F2          '+chr(24)+' '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice      Move Field Cursor'
@ 23,50 say ' PgDn          Esc'
@ 24,50 say 'Exit Screen  Abort Screen'
```

```
do getkey
ch:=input
```

```
set color to w/
@ 20, 47 SAY "RTP (micro sec):"
```

```
set color to /w
return
```

```
*****
*      RTPS Menu
*****
procedure RTPS
parameter ch
```

```
set color to w/
@ 21, 57 SAY "RTPS :"
```

```
set color to w/
@ 23,1 say '          F9          '+chr(24)+' '+chr(25)+' and '+chr(17)+'c
---->hr(196)+chr(217)
@ 24,1 say ' Toggle Choice      Move Field Cursor'
@ 23,50 say ' PgDn          Esc'
@ 24,50 say 'Exit Screen  Abort Screen'
```

```
do getkey
ch:=input
```

```
set color to w/
@ 21, 57 SAY "RTPS :"
```

```
set color to /w
return
```



```

*****
*           Course Gain Menu
*****
procedure cgain
parameter ch

set color to w/
@ 19, 9 SAY "Coarse"      :
set color to w/

@ 23,1 say '      F2           '+chr(24)+'', '+chr(25)+' and '+chr(17)+'c
....>hr(196)+chr(217)
@ 24,1 say ' Toggle Choice      Move Field Cursor'
@ 25,50 say ' PgDn      Esc'
@ 24,50 say 'Exit Screen  Abort Screen'

do getkey
ch=finput

set color to w/
@ 19, 9 SAY "Coarse"      :
set color to /w
return

*****
*           Next screen Menu
*****
procedure NextSc
set scoreboard off
parameter menuch
clear
@ 0,1 say "IAEA Plutonium Analysis System"
@ 1,1 say date()
@ 1,72 say time()
text

                Cicero Data Transfer

S) System Setup

I) Inspection Identification

F) File and Peak Information

T) Transfer Data from Analyzer

Q) Quit Data Entry and Exit

        Enter choice:

                Cicero Data Out Instructions
                .....
                1. Memory Range : Half 1
                2. Peripheral  : Computer
                3. Out. Mode   : Total

endtext
@ 2,22 to 4,60
@ 4,22 to 17,60
@ 4,22 say chr(195)
@ 4,60 say chr(180)
@ 24,7 say "Enter the appropriate key or use the arrow keys"
@ 24,55 say " then press enter"

row=15
col=42

```

```

selected = .f.
menuch=""
char = "S"
@ row,col say char
do while .not. selected
do funky
if finput=f1
do cichelp with 'C',char
endif
do selsc with menuch,selected,char
@ row,col say char
enddo
clear

set color to /w

return

```

```

*****
*---SELSC
* This procedure gets the choice in the menu
*****
procedure selc

para menuch,selected,char

if (finput=cr) .or. (finput = pgdn)
selected = .t.
return
else
do case
case finput = 83
char = "S"
menuch=1

case finput = 73
char = "I"
menuch=2

case finput = 70
char = "F"
menuch=3

case finput = 84
char = "T"
menuch=4

case finput = 81
char = "Q"
menuch=5

case finput = 5
menuch=menuch-1
if menuch<1
menuch=5
endif

case finput = 24
menuch=menuch+1
if menuch>5
menuch=1
endif

case finput = esc

```

```

        menuch=5
        selected = .t.
    endwhile
    endif
    if finput = 5 .or. finput = 24
        do case menuch=1
            case char = 'q'
                menuch=2
            case char = 'i'
                menuch=3
            case char = 'w'
                menuch=4
            case char = 'n'
                menuch=5
            case char = 'u'
                menuch=6
        endcase
        char = 'q'
    endif
    return
*****
*--ENTR
* calls getstrg and defines the needed public variables
*****
procedure entr
parameters str,r,c,maxcnt,flag,lib,win
do while flag=1
    do help
    do getstrg
enddo
return
*****
*--Dumpldsp
* strips off leading blanks
*****
procedure dumpldsp
parameters str
* ... strip off leading blanks
index=1
lenst=len(str)
stemp = substr(str,1,1)
do while stemp= ' '
    index=index+1
    lenst=lenstrg,2,lenst-1
    stemp=substr(str,1,1)
enddo
return
*****
procedure Clickhelp
parameters lib,win
do help
return
*****
procedure c1c0gettime
parameters str,r,c,flag,lib,win
do gettime
return
*****
procedure c1c0getnum
parameters val,r,c,maxcnt,flag,dec,prerr,lib,win
do while numgoods=f.

```

```

do getnum
if val<0
else
    numgood=f.
endif
numgood=t.
set color to w
if prerr=0
    if not numgood
        g r1,c say 'Re-enter'
    else
        g r1,c say ' '
    endif
endif
enddo
return
*****
procedure c1c0getdate
parameters date1,r1,c,flag,lib,win
do getdate
return

```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : Funkey
Revision : 1.0
Author : Russell Fleming
Date : 9/26/88
Language : Clipper Summer '87

Abstract : Gets only the ascii value for 4 character or
the code for up and down arrow keys.

Arguments :

Routines
Called :

Libraries
Used :

Revision History (Include date, revision I.D., and details)

```
finput=0
do while .not. ( finput=5, .or. finput=24, .or. finput<64 .and. finput<91, .or. finput<96 .and. finput<123 )
    finput = esc .or. finput = 13 .or. finput = pgdn .or. finput = f1
enddo
```

```
finput=inkey(.1)
@ 1,72 say time()
@ 1,1 say ''
```

```
enddo
if finput>96
    finput = finput - 32 && brings all the letters to upper case ascii code
endif
return
```

* End of code for FUNKEY.PRG

 International Atomic Energy Agency

Developed for : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

Developed by : COMPUchem, Inc.
 3099B Huntwood Avenue, Suite 107
 Hayward, CA. 94544
 (415)489-6514

This work supported by LLNL POW 9107605

Title : CHDEFAULT
 Revision : 1.0
 Author : Russell Fleming
 Date : 9/26/88
 Language : Clipper Summer '87

Abstract : Changes default settings of IAEA system. Change
 drive, path and IADEFALT settings.

 Arguments :

Routines
 Called :
 Libraries
 Used :

Revision History (Include date, revision I.D., and details)

```

updated=.f.
defempty=.f.
do while .t.
  clear
  @ 0,1 say "IAEA Plutonium Analysis System"
  @ 1,1 say date()
  @ 1,72 say time()
  set color to w/
text

```

Change Default Settings

- A) Change Analysis Defaults
- D) Change Default Drive and Path
- E) Exit This Menu

```

endtext
@ 6,24 to 8,63
@ 8,24 to 18,63
@ 8,24 say chr(195)
@ 8,63 say chr(180)
@ 16,30 say "Enter Choice:"
@ 24,7 say "<Enter the appropriate key or use the arrow keys"
@ 24,55 say " then press enter>"

```

**** Get menu choice and go to choice****

```

set cursor off
menuch=1
selected=.f.
char=""
@ 16,44 say char
do while .not. selected
  do funkey
  if finput=""
    do defhelp with 'D',ch."
  endif
  do seldef with menuch,selected,char
  @ 16,44 say char
enddo
set cursor on

```

***** case statement goes to case1) abundancies screen->batch sample info loop
 * case2) drive screen loop
 * case3) returns to iaea.prg

```

do case
case menuch=1
* .....
  iadef=drivepath+"iadefalt"
  if file("iadefalt.dbf")
    use iadeft

```

***** if the database is empty(no records) append blank record and
 ***** set flags to display errors

```

  if lastrec()=0
    updated=.f.
    defempty=.t.
    append blank
  else
    defempty=.f.
  endif
  else
    @ 22,0 clear
    @ 22,0 say "Error opening IADEFALT.DBF. Check default drive and
    ....>path settings"
    abort=.t.
    wait
    return
  endif
  * Bring in initial values
  xupdate= PUPDATE

```

```

xpu238 = pu238
xpu239 = pu239
xpu240 = pu240
xpu241 = pu241
xpu242 = pu242
xam241 = am241
xpu238err=pu238ERR
xpu239err=pu238ERR
xpu240err=pu238ERR
xpu241err=pu238ERR
xpu242err=pu238ERR
xam241err=am241err
xBATCH=batch
xCHAT=cmat
xCHAT2=cmat2
xCHAT3=cmat3
xWITCH=moitem
xHOC=mdc
xTPUMAS=tpumas
xTUMAS=tumes
xMETHOD=method
xenrich=enrich

set color to w/
clear

@ 1, 1 say date()
@ 1, 72 say time()

@ 0, 1 say "IAEA Plutonium Analysis System"
@ 4, 4 TO 6, 68
@ 6, 4 TO 21, 68
@ 6, 4 say chr(195)
@ 6, 68 say chr(180)

@ 5, 22 SAY "Change Default Abundances"

@ 7, 6 SAY "Pu Analysis Date:"
@ 7, 37 SAY "Am Analysis Date:"

@ 9, 6 SAY "Xpu-238 Abundance:"
@ 9, 37 SAY "Xpu-238 % Error:"
@ 11, 6 SAY "Xpu-239 Abundance:"
@ 11, 37 SAY "Xpu-239 % Error:"
@ 13, 6 SAY "Xpu-240 Abundance:"
@ 13, 37 SAY "Xpu-240 % Error:"
@ 15, 6 SAY "Xpu-241 Abundance:"
@ 15, 37 SAY "Xpu-241 % Error:"
@ 17, 6 SAY "Xpu-242 Abundance:"
@ 17, 37 SAY "Xpu-242 % Error:"
@ 19, 6 SAY "Xam-241 Abundance:"
@ 19, 37 SAY "Xam-241 % Error:"

if defempty
  set color to w/
  @ 22, 3 SAY "Default database is empty. Please enter all values on
  ....>d exit screen.>"
  set color to w/
endif

do inputbn
@ 7, 25 say xPUdate
@ 7, 55 say xAMdate
@ 9, 25 say xpu238 PICTURE "#####"
@ 11, 25 say xpu239 PICTURE "#####"
@ 13, 25 say xpu240 PICTURE "#####"

```

```

@ 15, 25 say xpu241 PICTURE "#####"
@ 17, 25 say xpu242 PICTURE "#####"
@ 19, 25 say xam241 PICTURE "#####"
@ 9, 53 say xpu238ERR PICTURE "#####"
@ 11, 53 say xpu239ERR PICTURE "#####"
@ 13, 53 say xpu240ERR PICTURE "#####"
@ 15, 53 say xpu241ERR PICTURE "#####"
@ 17, 53 say xpu242ERR PICTURE "#####"
@ 19, 53 say xam241ERR PICTURE "#####"

FIELDNUM=1
FLG=0
DOME = .F.
DO WHILE .NOT. DOME
  DO CASE
    CASE FIELDNUM=1
      do chgetdate with xpu2date,7,25,flg,'D','B'
    CASE FIELDNUM=2
      do chgetdate with xam2date,7,55,flg,'D','B'
    CASE FIELDNUM=3
      DO chgetnum WITH xpu238,9,25,10,flg,4,.,,'D','B'
    CASE FIELDNUM=4
      DO chgetnum WITH xpu239,11,25,10,flg,4,.,,'D','B'
    CASE FIELDNUM=5
      DO chgetnum WITH xpu240,13,25,10,flg,4,.,,'D','B'
    case fieldnum=6
      DO chgetnum WITH xpu241,15,25,10,flg,4,.,,'D','B'
    CASE FIELDNUM=7
      DO chgetnum WITH xpu242,17,25,10,flg,4,.,,'D','B'
    CASE FIELDNUM=8
      DO chgetnum WITH xam241,19,25,10,flg,4,.,,'D','B'
    CASE FIELDNUM=9
      DO chgetnum WITH xpu238err,9,53,8,flg,4,.,,'D','B'
    CASE FIELDNUM=10
      DO chgetnum WITH xpu239err,11,53,8,flg,4,.,,'D','B'
    CASE FIELDNUM=11
      DO chgetnum WITH xpu240err,13,53,8,flg,4,.,,'D','B'
    case fieldnum=12
      DO chgetnum WITH xpu241err,15,53,8,flg,4,.,,'D','B'
    CASE FIELDNUM=13
      DO chgetnum WITH xpu242err,17,53,8,flg,4,.,,'D','B'
    CASE FIELDNUM=14
      DO chgetnum WITH xam241err,19,53,8,flg,4,.,,'D','B'

  ENDCASE
  DO CASE
    CASE FLG=UP
      FIELDNUM=FIELDNUM+1
      IF FIELDNUM<14
        FIELDNUM=14
      ENDOIF
    CASE FLG=DOWN OR. FLG=ER
      FIELDNUM=FIELDNUM+1

```

```

IF FIELDNUM=14
  FIELDNUM=1
ENDIF
CASE FLG=ESC
  done=.t.
  case flg=pgdn
    done=.t.
    updated=.t.
    replace pdate with xupdate
    replace amdate with xamdate
    replace pu238 with xpu238
    replace pu239 with xpu239
    replace pu240 with xpu240
    replace pu241 with xpu241
    replace pu242 with xpu242
    replace am241 with xam241
    replace pu238err with xpu238err
    replace pu239err with xpu239err
    replace pu240err with xpu240err
    replace pu241err with xpu241err
    replace pu242err with xpu242err
    replace am241err with xam241err
    commit
    if file('status.ia')
      erase status.ia
    endif
  otherwise
    fieldnum=fieldnum+1
    if fieldnum=14
      fieldnum=1
    endif
  endcase
ENDCASE

```

```

ENODO

```

```

set color to w/
clear

```

```

@ 1, 1 say date()
@ 1,72 say time()
@ 0, 1 say "IAEA Plutonium Analysis System"
@ 3,25 say "Change Batch Sample Information"
@ 2, 3 to 4, 77
@ 11, 3 to 20, 77
@ 4, 3 to 11, 77
@ 4, 3 say chr(195)
@ 4,77 say chr(180)
@ 11, 3 say chr(195)
@ 11,77 say chr(180)

```

```

@ 5, 7 SAY "Batch Name:"
@ 5,39 SAY "Canning Materials:"
@ 7, 7 SAY "Number of Items:"
@ 9, 7 SAY "Material Description:"
@ 13, 7 SAY "Plutonium Mass:"
@ 13,44 SAY "Uranium Mass:"
@ 15,15 SAY "Plutonium Analysis Method:"
@ 17,20 SAY "Uranium Enrichment:"

```

```

if defempty .and. (.not. updated)

```

```

  set color to w/

```

```

  @ 22,3 SAY "Default database is empty. Please enter all values and exit
  -----> screen."

```

```

  set color to w/
endif

```

```

do inputban

```

```

@ 5, 19 say xBATCH
@ 5, 58 say xCHAT
@ 6, 58 say xCHAT2
@ 7, 58 say xCHAT3
@ 7, 24 say xNOITEM
@ 9, 29 say xMDC
@ 13, 23 say xTPUMAS
@ 13, 58 say xTUMAS
@ 15, 42 say xMETHOD
@ 17, 40 say xenrich

```

```

done = .f.

```

```

frum=1
do while .not. done

```

```

  do case

```

```

    case frum=1
      do chgetstr with xbatch,5,19,15,flg,'D','C'

```

```

    case frum=2
      do chgetstr with xcmat,5,58,15,flg,'D','C'

```

```

    case frum=3
      do chgetstr with xcmat2,6,58,15,flg,'D','C'

```

```

    case frum=4
      do chgetstr with xcmat3,7,58,15,flg,'D','C'

```

```

    case frum=5
      do chgetstr with xnoitem,7,24,5,flg,'D','C'

```

```

    case frum=6
      do chgetstr with xmdc,9,29,15,flg,'D','C'

```

```

    case frum=7
      do chgetstr with xtpumas,13,23,12,flg,3,.,t.,'D','C'

```

```

    case frum=8
      do chgetstr with xtumas,13,58,12,flg,3,.,t.,'D','C'

```

```

    case frum=9
      do chgetstr with xmethod,15,42,20,flg,'D','C'

```

```

    case frum=10
      do chgetstr with xenrich,17,40,10,flg,'D','C'

```

```

  endcase

```

```

  do case

```

```

    case flg=up
      frum=frum+1
      if frum=1
        frum=10
      endif

```

```

    endif

```

```

    case flg=down .or. flg=cr
      frum=frum+1
      if frum=10
        frum=1
      endif

```

```

    endif

```

```

    case flg=esc
      done=.t.

```

```

    case flg=pgdn

```

```

      replace batch with xBATCH

```

```

      replace cmat with xCHAT

```

```

      replace cmat2 with xCHAT2

```

```

      replace cmat3 with xCHAT3

```

```

      replace noitem with xNOITEM

```

```

      replace mdc with xMDC

```



```

*--SELDEF gets menu choice
*****
procedure seldef
para menuch,selected,char
if (finput = cr) .or. (finput=pgdn)
    selected = .t.
    return
else
    do case
        case finput = 65
            char = "A"
            menuch = 1
        case finput = 68
            char = "D"
            menuch = 2
        case finput = 69
            char = "E"
            menuch = 3
        case finput = 5
            menuch=menuch+1
            if menuch<1
                menuch=3
            endif
        case finput = 24
            menuch=menuch+1
            if menuch>3
                menuch=1
            endif
        case finput = esc
            menuch=3
            selected= .t.
    endcase
endif
if finput = 5 .or. finput = 24
    do case
        case menuch=1
            char="A"
        case menuch=2
            char="D"
        case menuch=3
            char="E"
    endcase
endif
return

*****
*...calls help routine with public parameters
*****
procedure defhelp
para lib,win
do help
return

*****
* ...CHGETNUM
* This procedure gets a number and then checks if the number is less than zero
* and prints and error if prerr(print error) is passed true

```

```

*****
procedure chgetnum
para val,r,c,maxcnt,flag,dec,prerr,lib,win

numgood=.f.
do while numgood=.f.
    do getnum
        if val<0
            numgood=.f.
        else
            numgood=.t.
        endif
        set color to w/
        if prerr
            if .not. numgood
                @ r+1,c say 'Re-enter'
            else
                @ r+1,c say ' '
            endif
        endif
    enddo
    return
*****
*...CHGETSTRG
* This procedure calls getstrg defining public variables
*****
procedure chgetstr
para strg,r,c,maxcnt,flag,lib,win

do getstrg
do while flag=f1
    do help
    do getstrg
enddo
return

*****
*...CHGETDATE
* This procedure calls getdate defining public variables
*****
procedure chgetdate
para date1,r,c,flag,lib,win
do getdate
return

```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : Gettime
Revision : 1.0
Author : Richard Wenning
Date : 9/26/88
Language : Clipper Summer '87

Abstract : This routine gets the time using editing features
similar to date entry. Full error checking is in effect.

Arguments : publicly defined-- strg,r,c,flag,lib,win

Routines
Called :

Libraries
Used :

Revision History (Include date,revision I.D., and details)

*para strg,r,c,flag,lib,win

set cursor on

home=1
end =6
pgup=18

hourstr =substr(strg,1,2)
minstr =substr(strg,4,2)
secstr =substr(strg,7,2)

hour =.t.
timegood=.f.

set color to /w
@ r,c say hourstr
@ r,c+2 say ':'
@ r,c+3 say minstr
@ r,c+5 say ':'
@ r,c+6 say secstr

hourcnt =1
mincnt =1
seccnt =1

```
do while .t.
do while hour
do get2num1 with hourstr,r,c,flag,hourcnt
do case
case flag=1
do help
loop
case flag=esc .or. flag=up .or. flag=down
min =.f.
sec =.f.
hour=.f.
exit
case hourcnt>2
mincnt=1
min =.t.
sec =.f.
hour =.f.
exit
case hourcnt<1 .or. flag=100 .or. flag=300 && home or backspace
hourcnt=1
loop
case flag=200 && end entered
sec =.t.
hour =.f.
min =.f.
seccnt=2
ip =.t.
exit
otherwise
min =.f.
hour =.f.
sec =.f.
exit
endcase
enddo
```

```
do while min
do get2num1 with minstr,r,c+3,flag,mincnt
do case
case flag=1
do help
loop
case flag=300 && backspace entered
hourstr=substr(hourstr,1,1)+''
hourcnt=2
hour =.t.
sec =.f.
min =.f.
@ r,c say hourstr
exit
case flag=esc .or. flag=up .or. flag=down
```

```

        exit
    case mincnt>2
        secnt=1
        sec =.t.
        min =.f.
        hour=.f.
        exit
    case mincnt<1
        hourcnt=2
        hour =.t.
        min =.f.
        sec =.f.
        exit
    case flg=100 && home entered
        hourcnt=1
        hour =.t.
        sec =.f.
        min =.f.
        exit
    case flg=200 && end entered
        sec =.t.
        min =.f.
        hour =.f.
        secnt=2
        exit
    otherwise
        min =.f.
        hour=.f.
        sec =.f.
        exit
    endcase
enddo
do while sec
do get2num1 with secstr,r,c+6,flg,secnt
do case
case flg=1
do help
loop
case flg=100 && home entered
        hourcnt=1
        hour =.t.
        sec =.f.
        min =.f.
        exit
    case flg=300 && backspace entered
        minstr=substr(minstr,1,1)+''
        min =.t.
        sec =.f.
        hour =.f.
        mincnt=2
        @ r,c+3 say minstr
        exit
    case secnt>2 .or. flg=esc
        exit
    case secnt<1
        mincnt=2
        hour =.t.
        min =.f.
        sec =.f.
        exit
    case flg=200 && end entered
        sec =.t.
        hour =.f.
        min =.f.
        secnt=2
        exit
enddo

```

```

        otherwise
            min =.f.
            hour=.f.
            sec =.f.
            exit
        endcase
    enddo
    if flg<esc .and. flg>100
        do checktm with hourstr,minstr,secstr,timegood
        if .not. timegood
            hourstr=space(2)
            minstr =space(2)
            secstr =space(2)
            @ r,c say hourstr+':'*minstr+':'*secstr
            hourcnt=1
            hour =.t.
            sec =.f.
            min =.f.
            loop
        endif
        @ r,c say hourstr+':'*minstr+':'*secstr
    endif
    if (secnt>2 .or. flg=up .or. flg=pgup .or. flg=down .or. flg=cr .or.;
        flg=pgdn) .and. timegood
        exit
    else
        if flg=esc
            exit
        endif
    endif
enddo
flag=flg

if flag=esc
    @ r,c say strg
    set cursor off
    return
else
    strg=hourstr+':'*minstr+':'*secstr
    if flag=pgup
        flag=up
    endif
    set cursor off
    return
endif
return

Procedure get2num1
para str,row,col,flg,cnt

tab =9
right =4
left =19
home =1
pgup =18
end =6
insertkey=22

insert =.f.
updateflag =.f.

@ row,col+cnt-1 say ''
ch=inkey(.1)

```

```

do while ch<>esc.and, ch<>up.and, ch<>down.and, ch<>pgdn;
.and, ch<>f1.and, ch<>cr.and, ch<>pgup
do case
case (ch=insertkey)
if (insert)
insert=.f.
else
insert=.t.
endif
case (ch=left)
cnt=cnt-1
if (cnt<1)
exit
endif
updateflag=.t.
case (ch=right)
cnt=cnt+1
if (cnt>2)
exit
endif
updateflag=.t.
case (ch=bs)
if cnt=1
flg=-300
return
else
str=substr(str,2,len(str)-1)
updateflag=.t.
cnt=1
endif
case (ch=del)
if cnt=1
str=substr(str,2,len(str)-1)
updateflag=.t.
else
str=substr(str,1,1)+''
updateflag=.t.
endif
case ch=home
flg=-100
return
case ch=end
flg=200
return
case cnt>2
exit
case ch=tab
cnt=3
exit
case (ch>47.and, ch<58)
if (insert)
if cnt=1
str=chr(ch)+substr(str,1,1)
else
str=substr(str,1,1)+chr(ch)
endif
else
if cnt=1
str=chr(ch)+substr(str,2,1)
else

```

```

str=substr(str,1,1)+chr(ch)
endif
endif
updateflag=.t.
cnt=cnt+1
endcase
if (updateflag)
ttt=2-len(str)
if (ttt>0)
a row,col say str+space(ttt)
else
a row,col say str
endif
a row,col+cnt-1 say ''
updateflag=.f.
endif
ch=inkey(.1)
enddo
flg=ch
set color to /w
return

```

```

procedure checktm
para hour,min,sec,timegood

```

```
timegood=.f.

```

```

if hour+min+sec=
timegood=.t.
endif
h=val(hour)
m=val(min)
s=val(sec)
if h>13 .or. m>59
return
endif
timegood=.t.
if h<10
hour='0'+str(h,1)
endif
if m<10
min='0'+str(m,1)
endif
if s<10
sec='0'+str(s,1)
endif
return

```

```

*
*
* International Atomic Energy Agency
*
*
*
* Developed for : Safeguards Technology Program
*                 Nuclear Chemistry Division
*                 Lawrence Livermore National Laboratory
*                 Livermore, CA. 94550
*
* Developed by :  COMPUCHEN, Inc.
*                 30998 Huntwood Avenue, Suite 107
*                 Hayward, CA. 94544
*                 (415)489-6514
*
* This work supported by LLNL POW 9107605
*
*
* Title       : Getnum
* Revision    : 1.0
* Author      : Richard Wenning
* Date       : 9/26/88
* Language    : Clipper Summer '87
*
*
* Abstract   : This routine perfectly mimics the clipper real number
*              GET statement.
*
*
* Arguments : publicly defined- val,r,c,maxcnt,dec,flag,lib,win
*
*
* Routines
* Called   :
*
* Libraries
* Used     :
*
*
* Revision History ( Include date,revision I.D., and details )
*
*
*
* parameters(publicly defined) val,r,c,maxcnt,dec,flag,lib,win
*
set cursor on

strval =str(val,maxcnt,dec)
wholestr=substr(strval,1,maxcnt-dec-1)
deastr =substr(strval,len(wholestr)+2)

whcnt =1
decent =1
pgup =18

```

```

transch=0

set color to /w
@ r,c          say wholestr
@ r,c+len(wholestr) say ' '
@ r,c+len(wholestr)+1 say deastr
@ r,c          say ' '

editdec=.f.  && -- flag set to edit the decimal field
flag=100  && -- flag passed to and from numentr for first position
*         && -- first time through.

ch=incr(1)
editdec=.f.  && -- flag set to show if the decimal field has been
*         && -- entered yet.

do while fig<=c .and. fig<pgdn .and. fig<up .and. fig<down .and. fig<cr;
    .and. fig<pgup
    do while .not. editdec
        do numentr with wholestr,r,c,len(wholestr),whcnt,fig,transch
        do case
            case fig=1
                do help
                loop
            case fig=600 && -- flag for home key which brings cursor to
                && -- beginning of whole number field.
                whcnt=1
                loop
            case fig=500 && -- flag for end key which brings cursor to
                && -- beginning of end of decimal field.
                decent=dec
                editdec=.t.
                exit
            case fig=700 && -- flag for insertion which carries over into
                && -- the decimal field. This is ignored when the
                && -- decimal editing loop and transch is simply lost.
                deastr=transch+substr(deastr,1,dec-1)
                @ r,c+len(wholestr)+1 say deastr
                @ r,c say wholestr
                whcnt=whcnt+1
                if whcnt>len(wholestr)
                    editdec=.t.
                    fig=700
                    exit
                endif
                loop
            case fig<up .or. fig<down
                exit
        endcase
        do case
            case whcnt>len(wholestr)
                editdec=.t.
                decent=1
                exit
            case whcnt<1
                whcnt=1
                loop
            otherwise
                if whcnt=1 .and. fig<up .and. fig<down .and. fig<cr .and.;
                    fig<pgdn .and. fig<pgup
                    && -- if at the first position zero out everything
                    deastr=space(dec)
                    @ r,c+len(wholestr)+1 say deastr
                    wholestr=space(len(wholestr)-1)+0'
                    else

```

```

if flg=-200
  && -- decimal was entered. Zero decimal, and jump to decimal
  && -- field
  @ r,c say wholestr
  decstr=space(dec)
  @ r,c len(wholestr)+1 say decstr
  loop
else
  if (flg=cr .or. flg=pgup .or. flg=down .or.
    flg=pgdn) .and. (.not. editeddec)
    && -- if any of these field exiting values or we have not
    && -- edited the decimal field, get the string value as far
    && -- as the user has typed.
    if whcnt<1
      temp=substr(wholestr,1,whcnt)
      wholestr=space(len(wholestr)-len(temp))+temp
      @ r,c say wholestr
      decstr=space(dec)
    endif
    exit
  else
    if editeddec .and. flg<=-300
      && -- If we have edited the decimal and a decimal
      && -- was not entered, exit and jump to decimal field
      wholestr=wholestr
      exit
    else
      if flg=-300
        && -- decimal was entered
        temp=substr(wholestr,1,whcnt-1)
        wholestr=space(len(wholestr)-len(temp))+temp
      else
        && -- decimal was not entered and we have not entered
        && -- the decimal field. Grab the whole number just
        && -- as much as the user has type and move to decimal
        && -- field.
        temp=substr(wholestr,1,whcnt-1)
        wholestr=space(len(wholestr)-len(temp))+temp
      endif
      exit
    endif
  endif
endif
endif
@ r,c say wholestr
editdec=t.
deccnt=1
endcase
enddo
if flg=up .or. flg=down .or. flg=pgdn .or. flg=esc .or. flg=cr .or.;
  flg=pgup
  exit
endif
do while editdec
  editdec=t.
  do numentr with decstr,r,c+len(wholestr)+1,dec,deccnt,flg,transch
  do case
    case flg=f1
      do help
    case flg=-600 && -- Flag for home key which brings cursor to
      && -- beginning of field.
      whcnt=1
      editdec=.f.
      exit
    case flg=-500 && -- Flag for end key which brings cursor to

```

```

*
  && -- beginning of end of decimal field.
  deccnt=dec
  loop
  case flg=-700 && -- Flag for insertion which carries over into
    && -- the decimal field. This is ignored when the
    && -- decimal editing loop and transch is simply lost.
    deccnt=deccnt+1
    @ r,c len(wholestr)+1 say decstr
  loop
endcase
do case
  case deccnt<1
    editdec=.f.
    whcnt=len(wholestr)
    exit
  case deccnt>dec
    flg=cr
    exit
  otherwise
    exit
endcase
enddo
if flg=up .or. flg=down .or. flg=pgdn .or. flg=esc .or. flg=cr .or.;
  flg=pgup
  exit
endif
enddo
flag=flg
do case
  case flag=esc
    val=val(strval)
  case flag=up .or. flag=down .or. flag=pgdn .or. flag=cr .or. flag=pgup
    pos=at('-',wholestr)
    if pos=len(wholestr)
      wholestr=space(len(wholestr)-2)+'0'
    endif
    val=val(wholestr+','+decstr)
  endcase
  @ r,c say str(val,maxcnt,dec)
set cursor off
return
*****
procedure numentr
para str,row,col,width,cnt,flg,transch

right =4
left =19
home =1
pgup =18
end =6
insertkey=22
space =32
period =46

insert =.f.
updateflag=.f.
@ row,col+cnt-1 say ''

do case
  case flg=-400
    ch=del

```

```

case flg=-700
  ch=insertkey
otherwise
  ch=inkey(.1)
endcase

do while ch<>esc .and. ch<>up .and. ch<>down .and. ch<>pgdn;
  .and. ch<>f1 .and. ch<>cr .and. ch<>pgup
do case
  case ch=insertkey
    if (insert)
      insert=.f.
    else
      insert=.t.
    endif
  case ch=left
    cnt=cnt-1
    if (cnt<1)
      exit
    endif
    updateflag=.t.
  case ch=right
    cnt=cnt+1
    if (cnt>width)
      exit
    endif
    updateflag=.t.
  case (ch=bs)
    do case
      case cnt=1
        flg=-400
        cnt=0
        return
      case cnt=2
        str=substr(str,2,len(str)-1)+''
        updateflag=.t.
        cnt=1
      otherwise
        str=substr(str,1,cnt-2)+substr(str,cnt,len(str)-cnt+1)+''
        updateflag=.t.
        cnt=cnt-1
    endcase
  case ch=home
    flg=-600
    return
  case ch=end
    flg=-500
    return
  case ch=period
    ch=-300
    exit
  case cnt>width .or. ch=-f1
    exit
  case (ch=del)
    if cnt=1
      str=substr(str,2,len(str)-1)+''
      updateflag=.t.
    else
      str=substr(str,1,cnt-1)+substr(str,cnt+1,len(str)-cnt)+''
      updateflag=.t.
    endif
  case (ch>47.and.ch<58) .or. ch=space .or. ch=asc('*') .or. ch=asc('~')

```

```

  if (insert)
    if cnt>1
      str=substr(str,1,cnt-1)+chr(ch)+substr(str,cnt)
      if (len(str)>width)
        temp=substr(str,1,width)
        transch=substr(str,len(temp)+1)
        str=temp
        flg=-700
        return
      endif
    else
      if flg=-100
        str=chr(ch)+space(width-1)
        flg=-200
        cnt=cnt+1
        return
      else
        str=chr(ch)+str
      endif
    endif
    if (len(str)>width)
      temp=substr(str,1,width)
      transch=substr(str,len(temp)+1)
      str=temp
      flg=-700
      return
    endif
    updateflag=.t.
    cnt=cnt+1
  else
    if cnt>1
      str=substr(str,1,cnt-1)+chr(ch)+substr(str,cnt+1,width-cnt)
    else
      if flg=-100
        str=chr(ch)+space(width-1)
        flg=-200
        cnt=cnt+1
        return
      else
        str=chr(ch)+substr(str,2,width-cnt)
      endif
    endif
    updateflag=.t.
    cnt=cnt+1
  endif
endcase

if (updateflag)
  ttt=width-len(str)
  if (ttt>0)
    @ row,col say space(ttt)+str
  else
    @ row,col say str
  endif
  @ row,col+cnt-1 say ''
  updateflag=.f.
endif
ch=inkey(.1)
enddo
flg=ch

set color to /w
return

```

```
*parameters lib,win
set cursor off
do case
```

```
  case lib='P'
    call fwclip with '-l=panel/'
    do case
      case win='F'
        call fwclip with "-W=fapi,nowait,row8,col11/"
      case win='P'
        call fwclip with "-W=pa,nowait,row8,col11/"
      case win='B'
        call fwclip with "-W=bsi,nowait,row8,col11/"
      case win='A'
        call fwclip with "-W=dpa,nowait,row9,col13/"
      case win='Q'
        call fwclip with "-W=aartmm,nowait,row9,col15/"
      case win='C'
        call fwclip with "-W=Cscreen,nowait/"
      case win='G'
        call fwclip with "-W=Gscreen,nowait/"
      case win='E'
        call fwclip with "-W=Escreen,nowait/"
    endcase
  case lib='M'
    call fwclip with '-l=mainwind/'
    do case
      case win='T'
        call fwclip with "-W=dtfca,nowait,row9,col1/"
      case win='A'
        call fwclip with "-W=daogrsod,nowait,row3,col1/"
      case win='C'
        call fwclip with "-W=qac,nowait,row5,col1/"
      case win='D'
        call fwclip with "-W=cds,nowait,row9,col1/"
      case win='Q'
        call fwclip with "-W=qtmarts,nowait,row9,col1/"
    endcase
  case lib='C'
    call fwclip with '-l=cictrans/'
    do case
      case win='S'
        call fwclip with "-W=ssyssetup,NOWAIT/"
      case win='I'
        call fwclip with "-W=inspecj,NOWAIT/"
      case win='F'
        call fwclip with "-W=ssinfo,NOWAIT/"
      case win='T'
        call fwclip with "-W=transfer,NOWAIT,ROW6,COL10/"
      case win='Q'
        call fwclip with "-W=adeae,NOWAIT,ROW6,COL10/"
      case win='SSI'
        call fwclip with "-W=SSI,nowait/"
      case win='B'
        call fwclip with "-W=SSSCREEN,nowait/"
      case win='C'
        call fwclip with "-W=INID,nowait/"
      case win='D'
        call fwclip with "-W=SAHSPEC,nowait/"
    endcase
  case lib='D'
    call fwclip with '-l=defchang/'
    do case
```

```
  case win='A'
    call fwclip with "-W=CAD,nowait,row8,col11/"
  case win='D'
    call fwclip with "-W=CDDAP,nowait/"
  case win='E'
    call fwclip with "-W=QTM,nowait,row10,col12/"
  case win='B'
    call fwclip with "-W=plabund,nowait/"
  case win='C'
    call fwclip with "-W=batch,nowait/"
  case win='F'
    call fwclip with "-W=drive,nowait/"
  case win='X'
    call fwclip with "-W=Kscreen,nowait/"
  endcase
```

```
  junk=0
  do while junk<>ff
    junk=inkey(.1)
  enddo
  call fwclip with '-c=all/'
  set cursor on
  return
```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEN, Inc.
3099B Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : Getdate
Revision : 1.0
Author : Richard Wenning
Date : 9/26/88
Language : Clipper Summer '87

Abstract : This routine perfectly mimics the clipper date GET
statement for the ansi date format. Full error
checking is in effect.

Arguments : (publically defined) date1,r,c,flag,lib,win

Routines
Called :
Libraries
Used :

Revision History (Include date,revision I.D., and details)

set cursor on

home=1
end =6
pgup=18

if date1=ctod(' . . . ')
*Check for passing of blank string
yearstr=' '
monstr=' '
daystr=' '

```
else
  temp =dtoc(date1)
  yearstr =substr(temp,1,2)
  monstr =substr(temp,4,2)
  daystr =substr(temp,7,2)
endif
```

```
year =-.t.
dategood=.f.
```

```
set color to /w
@ r,c say yearstr
@ r,c+2 say '. '
@ r,c+3 say monstr
@ r,c+5 say '. '
@ r,c+6 say daystr
```

```
yearcnt =1
monthcnt=1
daycnt =1
```

```
do while .t.
  do while year
    * && -- do while editing year field
    do get2num with yearstr,r,c,flag,yearcnt
    do case
      case flag=1
        do help
        loop
      case flag=ac .or. flag=up .or. flag=down
        month=.f.
        day=.f.
        year=.f.
        exit
      case yearcnt>2
        monthcnt=1
        month=.t.
        day=.f.
        year=.f.
        exit
      case yearcnt<1 .or. flag=100 .or. flag=300 && flag=home or backspace
        yearcnt=1
        loop
      case flag=200 && -- End was entered
        day=.t.
        year=.f.
        month=.f.
        daycnt=2
        exit
      otherwise
        month=.f.
        year=.f.
        day=.f.
        exit
    endcase
  enddo
  do while month
    do get2num with monstr,r,c+3,flag,monthcnt
    do case
      case flag=1
        do help
        loop
      case flag=300 && backspace key
        yearstr=substr(yearstr,1,1)+'
```



```

year=.t.
day=.f.
month=.f.
@ r,c say yearstr
exit
case flg=esc .or. flg=up .or. flg=down
day=.f.
exit
case monthcnt>2
daycnt=1
day=.t.
month=.f.
year=.f.
exit
case monthcnt<1
yearcnt=2
year=.t.
month=.f.
day=.f.
exit
case flg=-100 && home key entered
yearcnt=1
year=.t.
day=.f.
month=.f.
exit
case flg=-200 && -- End was entered
day=.t.
month=.f.
year=.f.
daycnt=2
exit
otherwise
month=.f.
year=.f.
day=.f.
exit
endcase
enddo
do while day
do get2num with daystr,r,c+6,flg,daycnt
do case
case flg=f1
do help
loop
case flg=-100 && homekey was entered
yearcnt=i
year=.t.
day=.f.
month=.f.
exit
case flg=-300 && backspace key was entered.
monstr=substr(monstr,1,i)*' '
month=.t.
day=.f.
year=.f.
monthcnt=2
@ r,c+3 say monstr
exit
case daycnt>2 .or. flg=esc
exit
case daycnt<1
monthcnt=2
year=.f.
month=.t.
day=.f.

```

```

exit
case flg=-200 && -- End was entered
day=.t.
year=.f.
month=.f.
daycnt=2
exit
otherwise
month=.f.
year=.f.
day=.f.
exit
endcase
enddo
if flg<esc .and. flg>-100
do checkdt with yearstr,monstr,daystr,dategood
if .not. dategood
yearstr=space(2)
monstr =space(2)
daystr =space(2)
@ r,c say yearstr+'.'+monstr+'.'+daystr
yearcnt=1
year=.t.
day=.f.
month=.f.
loop
endif
@ r,c say yearstr+'.'+monstr+'.'+daystr
endif
if ((daycnt>2 .or. flg=up .or. flg=pgup .or. flg=down .or. flg=scr .or.;
flg=pgdn) .and. dategood) .or.;
(yearstr+'.'+monstr+'.'+daystr)=' ' . . '
exit
else
if flg=esc
exit
endif
endif
enddo
flag=flg
if flag=esc
@ r,c say date1
set cursor off
return
else
date1=ctod(yearstr+'.'+monstr+'.'+daystr)
if flag=pgup
flag=up
endif
set cursor off
return
endif
return
*****

Procedure get2num
para str,row,col,flg,cnt
tab      =9
right    =4
left     =19
home     =1

```

```

pgup    =18
end      =6
insertkey=22

insert   =.f.
updateflag =.f.

@ row,col+cnt-1 say ""
ch=inkey(.1)

do while ch<>esc .and. ch<>up .and. ch<>down .and. ch<>pgdn;
    .and. ch<>f1 .and. ch<>cr .and. ch<>pgup
    do case
        case (ch=insertkey)
            if (insert)
                insert=.f.
            else
                insert=.t.
            endif
        case (ch=left)
            cnt=cnt-1
            if (cnt<1)
                exit
            endif
            updateflag=.t.
        case (ch=right)
            cnt=cnt+1
            if (cnt>2)
                exit
            endif
            updateflag=.t.
        case (ch=bs)
            if cnt=1
                flg=300
                return
            else
                str=substr(str,2,len(str)-1)
                updateflag=.t.
                cnt=1
            endif
        case (ch=del)
            if cnt=1
                str=substr(str,2,len(str)-1)
                updateflag=.t.
            else
                str=substr(str,1,1)+""
                updateflag=.t.
            endif
        case ch=home
            flg=100
            return
        case ch=end
            flg=200
            return
        case cnt>2
            exit
        case ch=tab
            cnt=3
            exit
    endcase
enddo

```

```

case (ch>47.and.ch<58)
    if (insert)
        if cnt=1
            str=chr(ch)*substr(str,1,1)
        else
            str=substr(str,1,1)+chr(ch)
        endif
    else
        if cnt=1
            str=chr(ch)*substr(str,2,1)
        else
            str=substr(str,1,1)+chr(ch)
        endif
    endif
    updateflag=.t.
    cnt=cnt+1
endcase

if (updateflag)
    ttt=2-len(str)
    if (ttt>0)
        @ row,col say str+space(ttt)
    else
        @ row,col say str
    endif
    @ row,col+cnt-1 say ""
    updateflag=.f.
endif
ch=inkey(.1)
enddo
flg=ch

set color to /w
return

procedure checkdt
para year,month,day,dategood

dategood=.f.

if (year+month+day)=*
    dategood=.t.
    return
endif

y=val(year)
m=val(month)
d=val(day)

if m<1 .or. y<1 .or. d<1
    dategood=.f.
    return
endif

do case
case m=1 .or. m=3 .or. m=5 .or. m=7 .or. m=8 .or. m=10 .or. m=12
    if d>31
        dategood=.f.
        return
    endif
case m=2
    if y%4=0 .and. d>30

```

```
        dategood=.f.  
        return  
    else  
        if d>28  
            dategood=.f.  
            return  
        endif  
    case m=4 .or. m=6 .or. m=8 .or. m=9 .or. m=11  
        if d>30  
            dategood=.f.  
            return  
        endif  
    otherwise  
        dategood=.f.  
        return  
endcase  
dategood=.t.  
if y<10  
    year='0'+str(y,1)  
endif  
if m<10  
    month='0'+str(m,1)  
endif  
if d<10  
    day='0'+str(d,1)  
endif  
return
```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
30998 Huntwood Avenue, Suite 107
Hayward, CA. 94544
(415)489-6514

This work supported by LLNL POW 9107605

Title : getstrg
Revision : 1.0
Author : Richard Wenning
Date : 9/26/88
Language : Clipper Summer '87

Abstract : Generic editing routine for strings

Arguments : publically defined strg,r,c,maxcnt,flag

Routines
Called :
Libraries
Used :

Revision History (Include date,revision I.D., and details)

*parameters strg,r,c,maxcnt,flag

set cursor on

```
insert=.f.
updateflag=.f.
oldstrg=strg
cnt=1
set color to /w
@ r,c say ''
```

ch=inkey(.1)

```
do while (ch<>esc .and. ch<>cr .and. ch<>pgup .and. ch<>pgdn .and.;
ch<>up .and. ch<>down .and. ch<>f9 .and. ch<>f10 .and.;
cnt<=maxcnt)
```

do case

```
case (ch=22)
if (insert)
insert=.f.
else
insert=.t.
endif
```

```
case (ch=19)
cnt=cnt-1
if (cnt<1)
cnt=1
endif
updateflag=.t.
```

```
case (ch=6)
cnt=cnt+1
if (cnt>maxcnt)
cnt=maxcnt
endif
updateflag=.t.
```

```
case (ch=be)
if cnt=2
strg=substr(strg,2,len(strg)-1)
updateflag=.t.
cnt=1
else
if cnt>2
strg=substr(strg,1,cnt-2)+substr(strg,cnt,len(strg)-cnt+1)
updateflag=.t.
cnt=cnt-1
endif
endif
```

```
case (ch=del)
if cnt=1
strg=substr(strg,2,len(strg)-1)
updateflag=.t.
else
strg=substr(strg,1,cnt-1)+substr(strg,cnt+1,len(strg)-cnt)
updateflag=.t.
endif
```

```
case (ch>51.and.ch<256)
```

```
if (insert)
@ 22,0 say 'INSERT'
if cnt>1
strg=substr(strg,1,cnt-1)+chr(c1)+substr(strg,cnt,maxcnt-cnt+1)
else
strg=chr(ch)+strg
endif
if (len(strg)>maxcnt)
strg=substr(strg,1,maxcnt)
endif
updateflag=.t.
cnt=cnt+1
else
```

```
@ 22,0 say ' '
if cnt>1
strg=substr(strg,1,cnt-1)+chr(ch)+substr(strg,cnt+1,maxcnt-cnt)
else
```

```

        endif
        updateflag=.t.
        cnt=cnt+1
    endif
    case ch=f1
        flag=f1
        set cursor off
        return
    endcase

    if (updateflag)
        ttt=maxcnt-len(strg)
        if (ttt>0)
            @ 23,1 say " ">"*strg+"<" ttt=""+str(ttt)
            @ r,c say strg+space(ttt)
        else
            @ 23,1 say " ">"*strg+"<"
            @ r,c say strg
        endif
        @ r,c+cnt-1 say ""
        updateflag=.f.
    endif
    ch=inkey(.1)
enddo

flag=ch
if cnt>=maxcnt .and. maxcnt>1
    flag=maxcnt
endif

set color to w/

if ch=esc
    stg=oldstrg
    set cursor off
    return
else
    if (ch=cr .or. ch=pgup .or. ch=pgdn .or. ch=up .or. ch=down .or.;
        ch=f9 .or. ch=f10 .or. cnt>=maxcnt)
        set cursor off
        return
    endif
endif

* if we get here, an error condition exists
flag=-99
return

```

International Atomic Energy Agency

Developed for : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

Developed by : COMPUCHEM, Inc.
 30998 Huntwood Avenue, Suite 107
 Hayward, CA. 94544
 (415)489-6514

This work supported by LLNL POW 9107605

Title : Getwhnum
 Revision : 1.0
 Author : Richard Wenning
 Date : 9/26/88
 Language : Clipper Summer '87

Abstract : Gets a whole numbers

Arguments : publicly defined-- val,r,c,maxcnt,flag,lib,win

Routines
 Called :
 Libraries
 Used :

Revision History (Include date,revision I.D., and details)

* parameters(publicly defined) val,r,c,maxcnt,flag,lib,win

set cursor on

strval =str(val,maxcnt,0)
 wholestr=strval

whcnt =1
 pgup =18

set color to /W
 @ r,c say wholestr
 @ r,c say "

flag=0
 ch=inkey(.1)

do while flag<>esc .and. flag<>pgdn .and. flag<>up .and. flag<>down .and. flag<>cr;
 .and. flag<>pgup

do whnumentr with wholestr,r,c,maxcnt,whcnt,flag

do case flag=fl
 case do help
 loop

endcase
 if flag=up .or. flag=down .or. flag=pgdn .or. flag=esc .or. flag=cr .or. ;
 flag=pgup
 exit
 endif

enddo

flag=flag

do case

case flag=esc

val=val(strval)

case flag=up .or. flag=down .or. flag=pgdn .or. flag=cr .or. flag=pgup

pos=at(1,1,wholestr)

if pos=len(wholestr)

val=0

else

val=val(wholestr)

endif

endcase

@ r,c say str(val,maxcnt)

set cursor off

return

procedure whnumentr
 para str,row,col,width,cnt,flag

right =4

left =19

home =1

pgup =18

end =6

insertkey=22

space =32

insert =.f.

updateflag =.f.

@ row,col+cnt-1 say "

ch=inkey(.1)

do while ch<>esc .and. ch<>up .and. ch<>down .and. ch<>pgdn;
 .and. ch<>fl .and. ch<>cr .and. ch<>pgup

do case ch=insertkey

if (insert)

insert=.f.

else

insert=.t.

endif

```

case ch=left
  cnt=cnt-1
  if (cnt<1)
    cnt=1
    ch=inkey(.1)
    loop
  endif
  updateflag=.t.

case ch=right
  cnt=cnt+1
  if (cnt>width)
    ch=cr
    exit
  endif
  updateflag=.t.

case (ch=bs)
  do case
    case cnt=1
      ch=inkey(.1)
      loop
    case cnt=2
      str=substr(str,2,len(str)-1)+ ' '
      updateflag=.t.
      cnt=1
    otherwise
      str=substr(str,1,cnt-2)+substr(str,cnt,len(str)-cnt+1)+ ' '
      updateflag=.t.
      cnt=cnt-1
    endcase
  case ch=home
    cnt=1
    @ row,col say ''
  case ch=end
    cnt=width
    @ row,col+cnt-1 say ''
  case cnt=width .or. ch=f1
    exit
  case (ch=del)
    if cnt=1
      str=substr(str,2,len(str)-1)+ ' '
      updateflag=.t.
    else
      str=substr(str,1,cnt-1)+substr(str,cnt+1,len(str)-cnt)+ ' '
      updateflag=.t.
    endif
  case (ch<47.and.ch<58) .or. ch=space .or. ch=asc('++') .or. ch=asc('..')
    if (insert)
      if cnt>1
        str=substr(str,1,cnt-1)+chr(ch)+substr(str,cnt,width-cnt)
      else
        str=chr(ch)+substr(str,1,width-1)
      endif
      updateflag=.t.
    else
      if cnt>1
        str=substr(str,1,cnt-1)+chr(ch)+substr(str,cnt+1,width-cnt)
      else
        if cnt=1 .and. (.not. updateflag)
          str=chr(ch)+space(width-1)
        else
          str=chr(ch)+substr(str,2,width-cnt)
        endif
      endif
      updateflag=.t.

```

```

      cnt=cnt+1
    endif
  endcase
  if (updateflag)
    ttt=width-len(str)
    if (ttt>0)
      @ row,col say space(ttt)+str
    else
      @ row,col say str
    endif
    @ row,col+cnt-1 say ''
    updateflag=.f.
  endif
  ch=inkey(.1)
enddo
if updateflag
  cnt=width
endif
flg=ch

set color to /w
return

```

File: IA.BAT

07/25/1989 08:30 -- Page: 001

```
echo off
flrun /m=6
idea
unflash
if exist mast.bat mast
```



```

#-----
# Make description file for IAEA fortran sources
#-----
#-----
# Macro Definitions
#-----
OPTIONS   = /c /Ox /4Yb /Ge /FPI
COMPILER  = fl
FORLIB    = iaeafor
OTHERLIBS = filesub+grafsub+acssub+dbssub+commsub
#-----
# Inference Rules
#-----
.for.obj:
$(COMPILER) $(OPTIONS) $*.FOR
lib $(FORLIB) --$*.obj;
#-----

```

```

ave.obj: ave.for
decay.obj: decay.for
c2jdat.obj: c2jdat.for
anresult.obj: anresult.for iaeadb.cmn
diaea.obj: diaea.for iaeadb.cmn
dsave.obj: dsave.for top.cmn
bwf.obj: bwf.for
bkgd.obj: bkgd.for
maxvel.obj: maxvel.for
stripl.obj: stripl.for
ncnts.obj: ncnts.for
rdsd.obj: rdsd.for
bkgrd.obj: bkgrd.for
ctail.obj: ctail.for
gfit2.obj: gfit2.for
pufity.obj: pufity.for
diags.obj: diags.for
more.obj: more.for
matinv.obj: matinv.for
idpmy.obj: idpmy.for
ttest.obj: ttest.for
dchdat.obj: dchdat.for

```

```

testbin.obj: testbin.for
rdbin.obj: rdbin.for
sppow.obj: sppow.for
getd.obj: getd.for iaea.cmn
strlen.obj: strlen.for
cicero.obj: cicero.for
$(COMPILER) $(OPTIONS) cicero.for
ia.obj: ia.for iaeadb.cmn iaea.cmn
$(COMPILER) $(OPTIONS) ia.for
dcheck.obj: dcheck.for
$(COMPILER) $(OPTIONS) dcheck.for
master.obj: master.for top.cmn
$(COMPILER) $(OPTIONS) master.for
master.exe: master.obj ia.obj cicero.obj dcheck.obj $(FORLIB).lib dbssub.lib
link /E master+ia+dcheck+cicero,,, $(FORLIB)+$(OTHERLIBS) /SE:256;

```

```
#-----
# make info file for IAEA dBASE programs
#-----

# Macro Definitions
#-----
COMPILER = clipper
OPTIONS = -m
PRGLIB = prgs
LIBRARIAN = lib
#-----

# Inference Rules
#-----
.prg.obj:
$(COMPILER) $*.PRG $(OPTIONS)
$(LIBRARIAN) $(PRGLIB) --$*.OBJ;
#-----

iaea.obj: iaea.prg
$(COMPILER) iaea $(OPTIONS)

checkia.obj: checkia.prg

chdefault.obj: chdefault.prg

funkey.obj: funkey.prg

cicdata.obj: cicdata.prg

getiadat.obj: getiadat.prg

dch.obj: dch.prg

enditall.obj: enditall.prg

decfrum.obj: decfrum.prg

inputban.obj: inputban.prg

getkey.obj: getkey.prg

getcool.obj: getcool.prg

iaea.exe: iaea.obj $(PRGLIB) $(LIBRARIAN)
link /E iaea,, $(PRGLIB) /SE:256;
```