UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

PROGRAM LIST OF ANALYSI,

A COMPUTER PROGRAM FOR CONTROL OF AN ISOMASS 54E THERMAL-IONIZATION, SINGLE-COLLECTOR MASS-SPECTROMETER

by

K. R. LUDWIG

OPEN-FILE REPORT 85-142

PROGRAM LIST OF ANALYST, A COMPUTER PROGRAM FOR CONTROL OF AN ISOMASS 54E THERMAL-IONIZATION, SINGLE-COLLECTOR MASS SPECTROMETER

INTRODUCTION

ANALYST is a computer program for controlling an Isomass 54E mass-spectrometer. The program is written in Hewlett-Packard EXTENDED BASIC 2.1 for a Series 200 Hewlett-Packard computer (HP-9836, HP-9816, or HP-9817). The program requires the Advanced Programming and Graphics Extensions binary programs and at least 1 megabyte of RAM. The required keyboard is either the standard HP-9836 keyboard or the Option 805 ASCII extended keyboard for the HP-9816/17 computers.

Documentation for the program is given in a separate U.S.G.S. Open-File Report (Ludwig, 1985), which explains in detail how to use the program, as well as the general philosophy behind it. This report is intended to provide the program listing and use as a reference for those wishing to excise or modify the various subprograms in **ANALYST**.

A copy of the current version of the program (debugging and modification is expected to be a long-term process) is available from the author on 5-1/4" or 3-1/2" disks on request.

REFERENCES

Ludwig, K.R., 1985, User's Manual for **ANALYST**, a Computer Program for Control of an Isomass 54E Thermal-Ionization, Single-Collector Mass-Spectrometer; U.S. Geological Survey Open-File Report 85-141, 89 p.

APPENDIX: PROGRAM LIST OF ANALYST

```
3 ! ***************************** ANALYST ************************************
6 ! MAR 22 1985  05:50 pM
9  ! HP-9836 Program for ISOMASS 54E control
12 ! Requires 400 Kbytes net memory, BASIC 2.0, AP2.1 and GRAPH2.1, Printer at 701
15  OPTION BASE 1
18 ! Printer_model=2631
21  Printer_model=2225 ! Thinkjet
24  Candump=(Printer_model()2631) ! can the printer take graphics-dumps?
27  IMAGE 4A,4Z
30  DUMP DEVICE IS 701 ! but disabled with 2631 printer
33  OUTPUT KBD;"SCRATCH KEY ";CHR$(255)&CHR$(88); ! scratch typing-aid softkey definitions
36  OFF KEY
39  CONTROL 2,1;0  ! cancel PRINTALL
42  CONTROL 1,4;0  ! cancel DISPLAY FUNCTIONS
45  PRINTER IS CRT
48  I_t=2  !  default integration time is 0.2 seconds
51         !  (if I_t=1, integration time is 1 sec; I_t=2 indicates .2 sec)
54  Stripchart=0          ! exit graphics-monitor mode
57  IF Run=0 THEN Run=1    ! in case of invalid program-start
60  OUTPUT KBD;CHR$(255)&CHR$(75);
63  PRINT USING "6/,K,/,12X,K,K";CHR$(131)&RPT$(" ",80),RPT$(" ",8)&" ***** THE PRINTER ISN'T RESPONDING ***** "&RPT$(" ",15),RPT$(
" ",83)&CHR$(128)
66  PRINT USING "2/,K";"PRESS THE "&CHR$(132)&"CLR I/O"&CHR$(128)&" KEY (UPPER RIGHT), CORRECT THE PROBLEM, THEN PRESS "&CHR$(132)&
"CONTINUE"&CHR$(128)
69  PRINTER IS 701;WIDTH (80)
72  PRINT USING "#,K";CHR$(27)&"&k0S"    ! normal print-size
75  PRINT USING "#,K";CHR$(27)&"&l8D"     ! 1/8" line spacing
78  IF Printer_model=2225 THEN
81    PRINT USING "#,K";CHR$(27)&CHR$(15) ! cancel bold print
84    PRINT USING "#,K";CHR$(27)&"&d@"   ! cancel underline
87  END IF
90  PRINTER IS CRT
93  OUTPUT KBD;CHR$(255)&CHR$(75);
96  IF Debug THEN          ! skip to BMC if not 1st RUN
99    IF Full_auto THEN    ! printout message if illegal full-auto exit
102     PRINTER IS Prtr(2)
105     PRINT USING "/,K,/";"+++++++ PAUSE exit from auto-running,  "&TIME$(TIMEDATE)&" +++++++"
108     Full_auto=0
111   END IF
114   GOTO Skey_bmc
117 END IF
120 !
123 !
126 Mag:IMAGE "$0FJ",4Z
129!
132  DIM Input$[60],Prompt$(18)[36],Response$(18)[50],U$[10],Tune(5,2),Temp$[50]
135  INTEGER Pr,Data_daly,Escape
138  INTEGER Uncentered_pks(8),Singlefocus(8),Triplefocus(8),Iso(8)
141  REAL Val(8),Use(18),Range(18,2),Flag(2),Resistor_const(6)
144   !
```

```
147  COM /General/ Z$[9],INTEGER Prtr(2),Subflag,Auto,Full_auto,Foc(8),I_t
150  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
153  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
156  COM /Magnet/ Mcoef(3),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
159  COM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel
162  COM /Filaments/ Filament(4),Fil$(4)[10],INTEGER Nfils,Filnum
165  COM /Data1/ Peak(8),Pk,Interfere(4,3),Normal(2),T00,INTEGER Data_iso(8),Data_collector,Inv,Bmc_out,Next_run,Spike
168  COM /Data2/ Acc(7),Lacc(7),Sigma$(7)[10],Delta$(7)[10],Aver(7),Last_aver(7),Last_ratio$(7)[7],Ratio$(7)[7]
171  COM /Data3/ Bkrds(4,12),Bkrd_var(12),Resdecay(80),INTEGER Sample,N,Nsets,Ref,Rf,Bkrd_rdgs(4,12),Bkrd_sigma(4,12)
174  COM /Spikedrun/ Spike$[8],Spikedrun_ratio$(3)[7],Spike_ratio(3),Natural_ratio(2),INTEGER Idif(3),Spikedrun_iso(3),Spkdrun_ref
177  COM /Keyboard/ Cn$[1],Ci$[1],Cb$[1],Cu$[1],Q$[1],Clear$[2]
180  !
183  COM /Protect1/ REAL Xtime,Yheight,Time0,Nuclide$(0:24)[6],Normal0(2),Bnorm,Debug,J0,Pk_offset$(5)[5],INTEGER Niso,Coarsebin(0:1
0),Run,Block,Magnet0(24),B
186  COM /Protect2/ Runclass$[10],Element$(20)[10],Collector$(0:1)[8],Zero_time,Noise_time,Run_name$(32)[50],Ffile,Bmc$[5],Runvar(32
,27),Runtype$(32)[8]
189  COM /Protect3/ Last_isotopes$[50],Numberofblocks,Wmin,Wmax,Fmax,Maxgrowth,Pkswitch_ratio(7)
192  COM /Protect4/ Sample_name$(16)[50],REAL Where,Start_time,Start_wait,Start_block,Good_blocks,Focnum,Preheated(32,2),Orig_fil(4)
,INTEGER Cf_iso,Estbar(16,2)
195  COM /Protect5/ From_iso,Lastfil,Fract_normal,Candump,Jump_dir,Dump_datagraf,Dump_stripchart,Valid_block,Verygood_blocks,Off_fil
(4),Type
198  COM /Protect6/ INTEGER Run_iso(32,8),Run_order(32),Order(8),Lastblock_iso(8),Mip,Pb_4678,Hv0
201   !
204  RESTORE System_data
207 !  Multiple Interface Address-Codes (Barrel-Motor Disabled)
210 !                  CUP-1    CUP-.2   DALY-1   DALY-.2
213 System_data:DATA $0MW0?:0,$0MW0?60,$0MW0):2,$0MW0)62
216 !
219 !  Multiple Interface Address-Codes (Barrel-Motor Enabled)
222 !      CUP-1    CUP-.2   DALY-1   DALY-.2
225  DATA $0MW0?:8,$0MW0?68,$0MW0)::,$0MW0)6:
228 !
231 !  Address-codes for filament-current supply (Sa-Cen,Sa-Side,PreH-Cen,Preh-Side)
234  DATA $0M0,$0M1,$0P0,$0P1
237  DATA BELOW,-SIDE,PEAK,+SIDE,ABOVE,CENT. FIL.,SIDE FIL.,PRE-H. CF,PRE-H. SF,1,701,.050,.0002,100,1E4, CUP , DALY ,bmc
240  DATA 54.21E-6,.7102,2.507E-6,4.917,0,0     ! DZC constants DINGBAT, DEC 06, 1983
243   ! the above DZC constants give the following memory after 1-sec. of beam (ppm/sec.):  14.8/1.5 - 7.3/2.5 - 4.1/3.5 - 2.6/4.5 -
 1.9/5.5 - 0.8/10.5
246   ! coarse-magnet range output values
249  DATA 7776,7775,7773,7767,7757,7737,7677,7577,7377,6777,5777
252  DATA 0,6.2E4,"123456789"," D",1,0,0,3,1,10,1,1,0,8,1.5
255  DATA 440,.1,620,.1,720,.1,540,.1,0,0
258  READ Mm$(*),Ff$(*),Pk_offset$(*),Fil$(*),Prtr(*),Noise(*),Mx(*),Collector$(*),Bmc$,Resistor_const(*),Coarsebin(*)
261  READ Run,Ions,Z$,D$,Sample,Block,Mu,J0,Numberofblocks,Nsets,Nfils,Filnum,Where,Xtime,Yheight,Tune(*)
264  Cn$=CHR$(128)  ! normal
267  Ci$=CHR$(129)  ! inverse
270  Cb$=CHR$(131)  ! blinking-inverse
273  Cu$=CHR$(132)  ! underlined
276  Clear$=CHR$(255)&CHR$(?5)  ! screen-clear
279  Q$=CHR$(34)  ! quote-symbol
282  MAT Filament= (0)
285  CALL Resdecay(Resistor_const(*),Resdecay(*))
288  OUTPUT KBD;Clear$;
291  PRINT USING "6/,28X,K,5/";"WELCOME TO  ANALYST "
294  FOR I=1 TO 4
297    Beep(0,0,0,0,Tune(*))
300  NEXT I
303  WAIT 1
306  OUTPUT KBD;Clear$;
309  PRINT USING "K,/,21X,K,/";" DIGITAL INTEGRATOR :  SET RESPONSE AT .03, TIMING AT  ON 1-IN1., GAIN AT 1,","INPUT AT 1.
```

```
"
312  PRINT USING "K,/,16X,K,/";" FA3 AMPLIFIER :  TURN ON, SET AMPS-FULL-SCALE AT 1E-5, GAIN AT 1,","RESPONSE TIME AT 30 mS
."
315  PRINT USING "K,/,8X,K,/";"TURN ON ELECTROMAGNET SUPPLY, MULTIPLIER SUPPLY, MOTOR CONTROL,","SYSTEM MONITOR, DIGITAL
INTEGRATOR."
318  PRINT "PRESS  BRANDENBURG  MAINS, THEN RESET BUTTONS."
321  PRINT USING "/,K,/";"TURN  FILAMENT KNOBS  TO RESET AND AUTO."
324  PRINT USING "K,/";"TURN  PROGRAMMABLE DEFLECTION UNIT  &  FOCUS UNIT  TO AUTO."
327  PRINT USING "K,/";"TURN  PROGRAMMABLE FOCUS UNIT  POWER AND OUTPUT TO ON, KV TO 7, 100V TO 8."
330  PRINT "PUT "&FNUn$("SYSTEM")&" DISK IN RIGHT-HAND DRIVE,   "&FNUn$("DATA")&" DISK IN LEFT-HAND DRIVE."
333  DISP Cn$&"(press "&FNUn$("CONTINUE")&" when ready)"
336  PAUSE
339  DISP
342  PRINT USING "18/,9X,K,9/";"IS THE BEAM VALVE OPEN ?    (press  CONTINUE  when checked)"
345  PAUSE
348  Check_elements(Element$(*),0)
351  DISP
354   !
357 Run:Auto=0
360  Full_auto=0
363  ASSIGN @Path1 TO "FOCUS:INTERNAL"
366  ENTER @Path1,1;Singlefocus(*) !        single-filament focus settings
369  ENTER @Path1,2;Triplefocus(*) !        triple-  "       "       "
372  ASSIGN @Path1 TO *
375  GOSUB Daly
378  Input$=REV$(DATE$(TIMEDATE)) ! check clock if date not in 1980's, => computer turned off
381  IF VAL(Input$[2])<>891 THEN CALL Clockset
384  GOSUB Element
387  GOSUB Autel
390  GOSUB Mfind
393  Shiftlabel(Stripchart,Sample_name$(*),Sample,Run,Full_auto,Nfils)
396  Debug=1
399  GOTO Bmc_1
402   !
405 Daly:CALL Daly_enable(Daly)
408  OFF KEY
411  IF Daly AND Debug AND NOT Ze(1) THEN
414    Zero(Filament(*),0,0,Daly,Magnet(L,2),Peak_inter,Mu,Mm$(*),Collector$(*))
417    IF Subflag THEN WAIT 3
420  END IF
423  RETURN
426 !
429 Element:Dispel(Element$(*))
432  A=0
435  INPUT "ENTER THE NUMBER OF THE ELEMENT     (or press  CONTINUE  to escape)",A
438  IF A=0 THEN Skey_bmc
441  Type=A
444  RETURN
447   !
450 Autel:Getel(Type,Runclass$,Nuclide$(*),Normal0(*),Bnorm,Run_name$(Sample),Niso,Rf,Hv0,Magnet0(*),Coarsebin(*),Ref)
453  IF Subflag THEN
456    IF Auto THEN Do_next_run
459    IF NOT Auto THEN Change_element
462  END IF
465  Miniso=0
468  Spike=0
471  Changed_el=1
474  IF NOT Auto THEN
477    LOOP
480      CALL Hv(Mm$(*),Hv,P,Mu,I_t)
```

```
483   EXIT IF P OR (ABS(Hv-Hv0)<4) ! HV must be within 4V of std value
486     PRINT USING "K,K,/";"ADJUST HV TO ",Hv0
489     Clunk
492   END LOOP
495   PRINT
498 END IF
501 Mu=0
504 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L<>0),Foc(*))
507 MAT Daly_ok= (0)
510 B=Magnet(L,2)
513 Jump_dir=1-2*(L=Niso)
516 RETURN
519  !
522  !
525 Bmc:B=Magnet(L,2)        ! Entrance-line for Beam-Monitor Condition
528 CONTROL KBD;0 ! caps lock OFF
531 OFF KNOB
534 OFF ERROR
537 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L<>0),Foc(*))
540 Stripchart=0
543 Bmc_1:PRINTER IS CRT       ! Alternate entrance-line
546 TRACE OFF
549 ON KEY 0 LABEL "0: CENTER PEAK" GOTO Calcent
552 ON KEY 1 LABEL "1: FOCUS BEAM" GOTO Calfoc
555 ON KEY 2 LABEL "2: CNTR BARREL" GOTO Calbar
558 ON KEY 3 LABEL "3: BEAM CHART" GOTO Chartmenu
561 ON KEY 4 LABEL "4: FIL. CURRS" GOTO Calfil
564 ON KEY 5 LABEL "5: CUP/DALY" GOTO Bmc_collchange
567 ON KEY 6 LABEL "6: SCAN MAGNET" GOTO Magscan
570 ON KEY 7 LABEL "7: CHANGE ELEM" GOTO Change_element
573 ON KEY 8 LABEL "8: NEW SAMPLE" GOTO Do_mfind
576 ON KEY 9 LABEL "9: TAKE DATA" GOTO Mandat
579 ON KEY 10 GOTO Magmenu
582 ON KEY 11 GOTO Opticsmenu
585 ON KEY 12 GOTO Barrelmenu
588 ON KEY 13 GOTO Collmenu
591 ON KEY 14 GOTO Ratiodata
594 ON KEY 15 GOTO Spikedata
597 ON KEY 16 GOTO Status
600 ON KEY 17 GOTO Calrunvar
603 ON KEY 18 GOTO Auto
606 ON KEY 19 GOTO Calfknob
609  !
612 ON KBD GOTO Keybranch                    !"Live" keyboard for branching
615 ON KNOB .15 GOSUB Fil_knob               ! KNOB rotated
618 C:LOOP
621   OUTPUT 8 USING "4A,4Z";"$OMV",Magnet(L,1)
624   IF L THEN CALL Enter_beam(0,Mv,L,I_t,Pr,0,0,1)! non-187
627   IF L=0 THEN CALL Enter_beam(0,Mv,L,I_t,Pr)    ! Re-187
630   IF Pr=3 AND L>0 THEN ! if beam>10 volts, jump to next isotope (if L<>0)
633     Over_iso=1+Over_iso
636     IF Over_iso>Niso THEN
639       Enter_beam(0,Mv,L,I_t,Pr)
642     ELSE
645       PRINT TABXY(20,18);CHR$(131);" ***";Magnet(L,1);" PEAK IS >10 VOLTS *** "&CHR$(128)
648   ! new isotope is next one in direction of last bmc-requested peakjump
651       L=L+Jump_dir
654       IF L>Niso THEN L=1
657       IF L=0 THEN L=Niso
660       B=Magnet(L,2)
```

```
663       OUTPUT 8 USING Mag;B
666       Beep(3000,.02,.001,15)
669     END IF
672   ELSE
675     Over_iso=0
678   END IF
681   IF NOT Mu THEN Yd=DROUND(Mv,1+(Mv>1)+(Mv>=10)+(Mv>=1000))! rounded beamsize
684   IF Mu THEN Yd=DROUND(Mv,1+(Mv>.01)+(Mv>.5)+(Mv>10))
687   Running_time=(TIMEDATE-Time0)/60! time since sample rotated into position(min.)
690   IF Stripchart THEN
693     IF Running_time>Tmax THEN
696       IF Dump_stripchart AND Candump THEN DUMP GRAPHICS
699       GOTO Monitor
702     END IF
705     IF (NOT Logmon) THEN DRAW Running_time,Mv! (graphic beam-monitor)
708     IF Logmon AND (Mv>0) THEN DRAW Running_time,LGT(ABS(Mv)+(NOT Mv))! ditto but logarithmic
711   END IF
714   DISP USING 720;Ci$,Magnet(L,1),Cn$,Nuclide$(L),Pk_offset$(JO),B,Filament(1),Filament(2),UAL$(Yd),Collector$(Mu)
717 END LOOP
720 IMAGE A," ",3D," ",A,2X,6A,3X,5A,X,4D,3X,"CF=",D.3D,4X,"SF=",D.3D,4X,5A,X,"mU",4X,K,4X,"bmc"
723  !
726  ! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
729  ! **************************************************************************
732 !
735 Fil_knob:STATUS 2,10;K  ! KNOB rotated; K=1 if shift-key pressed also,
738                         ! 2 if control-key, 3 if both
741 IF K<2 OR (K=3 AND Nfils<2) THEN RETURN
744 Curr_change=INT(1000.*KNOBX/2000)/1000
747 Filnum=K-1 ! change center-fil if CONTROL-KNOB, side-fil if CTL-SHIFT KNOB
750 F=Filament(Filnum)+Curr_change
753 F=(F>0)*(F*(F<=7)+7*(F>7))
756 Filament(Filnum)=F
759 OUTPUT 8 USING "4A,4Z";Ff$(0,Filnum),FNF(F)
762 H=(F<.4)*100+(F).4)*F*800
765 IF H>4000 THEN H=4000
768 IF TIMEDATE-Tknob>.15 THEN BEEP H,.02
771 Tknob=TIMEDATE ! don't beep any more often than every .2 sec
774 MAT Daly_ok= (0)
777 Daly_ok(L)=Mu
780 RETURN
783 !
786 Softkeys:   ! put softkey-branches here
789 Calcent:Center_peak(1,L)
792 GOTO Bmc_1
795 Calfoc:Focus(4,1,1,Focnum)
798 Stripchart=0
801 GOTO Bmc_1
804 Calbar:Center_barrel(Sample)
807 IF Subflag=1 THEN
810   GRAPHICS OFF
813   PRINT USING "18/,4X,K,6/";FNB1$("********** CAN'T KEEP FILAMENT-CONTACTS FOR THIS SAMPLE **********")
816   Superclunk
819 END IF
822 Stripchart=0
825 GOTO Bmc_1
828 Magscan:GOSUB Clearall
831 IF Miniso=0 THEN
834   Miniso=Magnet(1,1)
837   Maxiso=Magnet(Niso,1)
840   Mass_speed=.2
```

```
843    Max_pk=10000
846    Logscan=0
849  END IF
852  Scan(Stripchart,Miniso,Maxiso,Max_pk,Mass_speed,Logscan,Coarsebin(*),Niso,I_t)
855  GOTO Bmc
858  !
861 Caltest:GOSUB Clearall
864  Contact_test(Filament(*),Time0,E,Sample_name$(*),Run_name$(*),Candump,Estbar(*),Sample,Nfils)
867  ON 1+E GOTO Bmc_1,Skey_bmc
870  !
873 Do_mfind:GOSUB Mfind
876  GOTO Bmc_1
879  !
882 Bmc_collchange:GOSUB Collchange
885  GOTO Bmc_1
888 !
891 Collchange: ! Toggle from Faraday Cup to Daly Detector or vice versa
894  IF Mu THEN   ! go from Daly to Faraday Cup
897    PRINT TABXY(1,18);FNUn$(" ON FARADAY CUP ");RPT$(" ",60)
900    BEEP 200,.1   ! Switch from Daly to  Faraday Cup as collector
903    Mu=0
906    Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),0,I_t,Coarsemag(L()0),Foc(*))
909    WAIT 5
912  ELSE   ! Switch from Faraday Cup to Daly
915    A=Daly_ok(L)
918    Mtest   !   check if Daly can be used for this peak, use if OK
921    IF A THEN WAIT .2
924  END IF
927  RETURN
930  !
933 Calfil:GOSUB Clearall  ! change filament-current
936  DATA FILAMENT-NUMBER,"(1=center-sample,  2=side-sample"," 3=center-preheat, 4=side-preheat)","",NEW CURRENT (amperes),"","",RAT
E (milliamps/second)
939  DATA 0,0,0,0,1,4,0,0,0,7,0,0,0,0,.01,1E6
942  RESTORE 936
945  REDIM Prompt$(8),Response$(8),Use(8),Range(8,2)
948  READ Prompt$(*),Range(*)
951  MAT Use= (0)
954  Use(3)=1
957  Use(5)=1
960  Use(8)=1
963  MAT Response$= ("")
966  ! pass last filament# to menu
969  IF Lastfil=0 THEN Response$(3)=VAL$(Filnum)
972  IF Lastfil THEN Response$(3)=VAL$(Lastfil)
975  IF Nfils=1 THEN
978    Response$(3)="1"
981    Prompt$(6)="(present CF current is "&VAL$(Filament(1))&" amps)"
984  END IF
987  IF Nfils=2 THEN Prompt$(6)="(present CF="&VAL$(Filament(1))&", SF="&VAL$(Filament(2))&")"
990  Response$(5)="??"
993  IF Lastrate=0 THEN Response$(8)="10"
996  IF Lastrate THEN Response$(8)=VAL$(Lastrate)
999  Form(Prompt$(*),Response$(*),Use(*),Range(*),8,3*(Nfils=2)+5*(Nfils=1),E)
1002 IF E THEN Skey_bmc
1005 Lastfil=VAL(Response$(3))
1008 Lastrate=VAL(Response$(8))
1011 CALL Filament(Lastrate,VAL(Response$(5)),.5,(Lastfil))
1014 Whoop
1017 OUTPUT KBD;Clear$;
```

```
1020 IF F<3 THEN Skey_bmc
1023 PRINT "PREHEAT: CF=";Filament(3);"      ";"SF=";Filament(4);CHR$(13)&CHR$(10)
1026 GOTO Bmc
1029 !
1032 Calfknob:CALL Filamentknob(Stripchart,Logmon,TimeO,Collector$(*),I_t,L,Magnet(L,1))
1035 ON 1+Stripchart GOTO Skey_bmc,Bmc_1
1038!
1041 Magmenu:! General menu for magnet-related functions
1044 GOSUB Clearall
1047 REDIM Response$(10)
1050 MAT Response$= ("")
1053 Response$(1)="SCAN MAGNET (AUTO)"
1056 Response$(3)="CHECK PEAKSHAPE"
1059 Response$(4)="DISPLAY MAGNET & OTHER ELEMENT-DATA"
1062 Response$(5)="CHANGE ELEMENTS (increments RUN#)"
1065 Response$(2)="SCAN MAGNET USING "&FNUn$("KNOB")
1068 Response$(6)="DEFINE DATA FOR A NEW ELEMENT"
1071 Response$(7)="ADJUST MAGNET-VALUES FOR SLIGHT DRIFT"
1074 Response$(10)="RETURN TO BMC"
1077 CALL Keymenu(Response$(*))
1080 ON KEY 0 LABEL "0: SCAN MAGNET" GOTO Magscan
1083 ON KEY 2 LABEL "2: PEAK SHAPE" GOTO Calshape
1086 ON KEY 3 LABEL "3: DISP VALUES" GOTO Element_print
1089 ON KEY 4 LABEL "4: CHANGE EL" GOTO Change_element
1092 ON KEY 1 LABEL "1: MAGNET KNOB" GOTO Magnet_knob
1095 ON KEY 5 LABEL "5: NEW ELEMENT" GOTO Cal_newel
1098 ON KEY 6 LABEL "6: DRIFTADJUST" GOTO Cal_driftadjust
1101 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1104 GOTO 1104
1107!
1110 Calshape:M=Mu !     do a graphic check of peakshape
1113 T=I_t
1116 GOSUB Clearall
1119 PRINT USING "2/,K,3/";" GRAPHICS PEAK-SHAPE CHECK: "
1122 Center_peak(1,L)
1125 OUTPUT 8;Mm$(M,T) ! restore original integration-time
1128 IF M<Mu THEN WAIT 6
1131 Mu=M
1134 I_t=T
1137 IF Subflag=4 THEN Skey_bmc
1140 CALL Shape(Mu,Nuclide$(*),I_t)
1143 OUTPUT KBD;Clear$;
1146 GOTO Bmc
1149!
1152 Magnet_knob:CALL Magknob(Coarse,B,Coarsebin(*),Mu)
1155 GOTO Skey_bmc
1158!
1161 Change_element:GOSUB Element
1164 GOSUB Autel
1167 GOTO Bmc
1170 !
1173 Cal_newel:CALL Newel(Type,Interfere(*),Normal(*),Nuclide$(*),Element$(*),Runclass$,E,Coarsebin(*),Niso,Ref,Rf,Inv,HvO)
1176 IF NOT E THEN GOSUB Autel
1179 ON 1+E GOTO Skey_bmc,Change_element
1182 !
1185 Cal_driftadjust:Drift_adjust(Type,Runclass$,Nuclide$(*),NormalO(*),NormO,Na$,Niso,Rf,HvO,MagnetO(*),Coarsebin(*),Ref)
1188 GOTO Skey_bmc
1191  !
1194 Opticsmenu:! Menu for functions related to ion optics
1197 StripchartO=Stripchart
```

```
1200 GOSUB Clearall
1203 REDIM Response$(10)
1206 MAT Response$= ("")
1209 Response$(1)="AUTOFOCUS ION-OPTICS"
1212 Response$(2)="FOCUS ION-OPTICS USING KNOB"
1215 Response$(3)="SCAN FOCUS-POTENTIALS"
1218 Response$(4)="DISPLAY FOCUS VALUES"
1221 Response$(5)="RESTORE FOCUS USING STANDARD FOCUS-VALUES"
1224 Response$(6)="DEFINE & STORE PRESENT FOCUS-VALUES AS STANDARD"
1227 Response$(7)="QUERY HIGH-VOLTAGE"
1230 Response$(8)="TYPE IN NEW FOCUS-SETTINGS"
1233 Response$(9)="COMPLETE BEAM TUNE-UP"
1236 Response$(10)="RETURN TO BMC"
1239 CALL Keymenu(Response$(*))
1242 ON KEY 0 LABEL "0: AUTO FOCUS" GOTO Calfoc
1245 ON KEY 1 LABEL "1: MANUAL FOC." GOTO Calmfocus
1248 ON KEY 2 LABEL "2: SCAN FOCUS" GOTO Calfocscan
1251 ON KEY 3 LABEL "3: DISP VALUES" GOTO Focprint
1254 ON KEY 4 LABEL "4: USE STD" GOTO Do_stdfocus
1257 ON KEY 5 LABEL "5: DEFINE STD" GOTO Defstd
1260 ON KEY 6 LABEL "6: HV?" GOTO Calhv
1263 ON KEY 7 LABEL "7: TYPE IN FOC" GOTO Calfocshift
1266 ON KEY 8 LABEL "8: BEAM TUNEUP" GOTO 2454
1269 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1272 ON KEY 10 GOTO Call_hvchange
1275 PRINT USING "2/,4X,K";"10 -- Change Standard-HV Values for Several Elements"
1278 GOTO 1278
1281!
1284 Calfocshift:CALL Focshift(Foc(*)) ! change focus settings to arbitrary values
1287 GOTO Skey_bmc
1290 !
1293 Calmfocus:GOSUB Clearall  ! focus the ion optics "manually" using KNOB
1296 CALL Mfocus(Stripchart0,Logmon,Time0,I_t,Foc(*),Z$)
1299 Stripchart=Stripchart0
1302 ON 1+Stripchart GOTO Skey_bmc,Bmc_1
1305 !
1308 Calfocscan:GOSUB Clearall   ! scan focus-settings
1311 Focscan(Foc(*),I_t)
1314 GOTO Skey_bmc
1317 !
1320 Call_hvchange:Change_hv
1323 GOTO Skey_bmc
1326 !
1329 Do_stdfocus:OUTPUT KBD;Clear$;
1332 GOSUB Stdfocus
1335 PRINT USING "/,K";"press "&FNH$("CONTINUE")&" when ready."
1338 PAUSE
1341 GOTO Skey_bmc
1344 !
1347 Defstd:ON ERROR GOTO Bad_store ! store current focus-settings on disk
1350 ASSIGN @Path1 TO "FOCUS:INTERNAL"
1353 OUTPUT @Path1,Nfils;Foc(*)
1356 PRINT USING "18/,K,4/";"CURRENT FOCUS-SETTINGS NOW STORED ON DISK AS STANDARD SETTINGS."
1359 PRINT USING "K,2/,8(3D,3X),/,8(4D,2X),/";"FOCUSING VALUES:",1,2,3,4,5,6,7,8,Foc(*)
1362 IF Nfils=1 THEN MAT Singlefocus= Foc
1365 IF Nfils=2 THEN MAT Triplefocus= Foc
1368 GOTO Bmc_1
1371 !
1374 Barrelmenu:! Menu for barrel-related functions
1377 GOSUB Clearall
```

```
1380 REDIM Response$(10)
1383 MAT Response$= ("")
1386 Response$(1)="AUTO-ADJUST BARREL-POSITION FOR BEST BEAM"
1389 Response$(2)="MANUAL BARREL-ADJUST"
1392 Response$(3)="TEST CONTACTS FOR ALL SAMPLES (turns fils OFF)"
1395 Response$(10)="RETURN TO BMC"
1398 CALL Keymenu(Response$(*))
1401 ON KEY 0 LABEL "0: AUTO-ADJUST" GOTO Calbar
1404 ON KEY 1 LABEL "1: MAN.-ADJUST" GOTO Calmbar
1407 ON KEY 2 LABEL "2: CALIBRATE" GOTO Caltest
1410 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1413 GOTO 1413
1416!
1419 Calmbar:GOSUB Clearkey
1422 CALL Mbar(Mm$(*),Barrel_position,Barrel_pos0,Min_barrel,Max_barrel,I_t)
1425 GOTO Skey_bmc
1428!
1431 Ratiodata:! Menu for accessing isotope-ratio data on disk
1434 GOSUB Clearall
1437 REDIM Response$(10)
1440 MAT Response$= ("")
1443 Response$(1)="START TAKING ISOTOPE-RATIO DATA"
1446 Response$(2)="ENTER NAMES FOR ALL SAMPLES IN BARREL"
1449 Response$(3)="PRINTOUT SUMMARY OF RUN(S)"
1452 Response$(4)="DISPLAY SUMMARY OF RUN(S)"
1455 Response$(5)="CALCULATE WEIGHTED AVERAGES OF RATIOS FOR A RUN"
1458 Response$(6)="LOCATE RUN-DATA ON A DATA-DISK"
1461 Response$(7)="SHOW A LIST OF CURRENTLY-DEFINED SAMPLE-NAMES"
1464 Response$(10)="RETURN TO BMC"
1467 CALL Keymenu(Response$(*))
1470 ON KEY 0 LABEL "0: TAKE DATA" GOTO Mandat
1473 ON KEY 1 LABEL "1: ENTER NAMES" GOTO Calnames
1476 ON KEY 2 LABEL "2: PRINT-RES" GOTO Prt_result
1479 ON KEY 3 LABEL "3: CRT-RESULTS" GOTO Crt_result
1482 ON KEY 4 LABEL "4: WTD AVERAGE" GOTO Calav
1485 ON KEY 5 LABEL "5: LOCATE DATA" GOTO Catalog
1488 ON KEY 6 LABEL "6: SHOW NAMES" GOTO Cal_shownames
1491 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1494 GOTO 1494
1497!
1500 Calav:CALL Average(0,Normal(*),Candump,E,(Run),Prtr(*))
1503 ON 1+E GOTO Bmc_1,Skey_bmc
1506 Catalog:CALL Cat(Prtr(*))  !        printout catalog of data on disk
1509 GOTO Skey_bmc
1512 Prt_result:P=2
1515 GOSUB Result
1518 GOTO Skey_bmc
1521 Crt_result:P=1
1524 GOSUB Result
1527 GOTO Bmc_1
1530 Calnames:CALL Name(Sample_name$(*),Estbar(*))
1533 GOTO Skey_bmc
1536 Cal_shownames:Show_names(Sample_name$(*))
1539 GOTO Bmc_1
1542!
1545 Spikedata:! Menu for accessing spike-related functions
1548 GOSUB Clearall
1551 REDIM Response$(10)
1554 MAT Response$= ("")
1557 GRAPHICS OFF
```

```
1560 Response$(1)="DISPLAY SPIKE-DATA STORED ON DISK"
1563 Response$(2)="DEFINE A NEW NORMALIZABLE-ELEMENT SPIKE"
1566 Response$(10)="RETURN TO BMC"
1569 CALL Keymenu(Response$(*))
1572 ON KEY 0 LABEL "0: DISPLAY" GOTO Showspikes
1575 ON KEY 1 LABEL "1: DEFINE NEW" GOTO Calnewspike
1578 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1581 GOTO 1581
1584!
1587 Showspikes:CALL Whichspike(0,0,Subflag)   ! show #s & names of spikes defined on disk
1590 ON 1+Subflag GOTO Bmc_1,Skey_bmc
1593!
1596 Calnewspike:CALL Nspike
1599 GOTO Skey_bmc
1602!
1605!
1608 Status:  ! Menu for accessing mass-spec status functions
1611 GOSUB Clearall
1614 REDIM Response$(10)
1617 MAT Response$= ("")
1620 Response$(1)="SOURCE/TUBE PRESSURE (SINGLE QUERY)"
1623 Response$(2)="SOURCE/TUBE PRESSURE (GRAPHICS)"
1626 Response$(3)="HIGH VOLTAGE"
1629 Response$(4)="QUERY TIME/DATE"
1632 Response$(5)="SET TIME/DATE"
1635 Response$(6)="TEST CONTACTS FOR BARREL# "&VAL$(Sample)
1638 Response$(7)="TEST CONTACTS FOR ALL SAMPLES (turns fils OFF)"
1641 Response$(8)="MAGNET & RUNNING DATA (current)"
1644 Response$(9)="FOCUS VALUES (current)"
1647 Response$(10)="RETURN TO BMC"
1650 CALL Keymenu(Response$(*))
1653 ON KEY 0 LABEL "  PRESSURE-1" GOTO Calpres
1656 ON KEY 1 LABEL "PRESSURE-GRAPH" GOTO Calpgraph
1659 ON KEY 2 LABEL "  HI VOLTAGE" GOTO Calhv
1662 ON KEY 3 LABEL "  READ CLOCK" GOSUB Querytime
1665 ON KEY 4 LABEL "   SET CLOCK" GOSUB Calset
1668 ON KEY 5 LABEL "  CONTACTS-1" GOTO Calflag
1671 ON KEY 6 LABEL " CONTACTS-16" GOTO Caltest
1674 ON KEY 7 LABEL "ELEMENT VALUES" GOTO Element_print
1677 ON KEY 8 LABEL " FOCUS VALUES" GOTO Focprint
1680 ON KEY 9 LABEL "    ESCAPE" GOTO Skey_bmc
1683 GOTO 1683
1686!
1689 Calpgraph:CALL Press_graph(Mm$(*),Candump)
1692 GOTO Bmc
1695 Calset:Clockset
1698 RETURN
1701 Querytime:PRINT USING "/,K,5X,K,/";FNClock_12$(TIME$(TIMEDATE)),DATE$(TIMEDATE)
1704 RETURN
1707 Calpres:GOSUB Clearall
1710 CALL Pressure(Mu,Mm$(*),(I_t),1,1)
1713 GOTO Bmc_1
1716 Calhv:GOSUB Clearall   ! query accelerating voltage
1719 CALL Hv(Mm$(*),Hv,0,Mu,I_t)         ! query accelerating voltage
1722 GOTO Bmc_1
1725 Element_print:Disp_elvals(Runclass$,Nuclide$(*),Normal(*),Interfere(*),Coarse,Ref,Hv0,Magnet(*),Niso,Inv)
1728 GOTO Bmc_1
1731 Focprint:GOSUB Clearall
1734 PRINT USING "K,2/,8(3D,3X),/,8(4D,2X),/";"FOCUSING VALUES:",1,2,3,4,5,6,7,8,Foc(*)
1737 GOTO Bmc_1
```

```
1740 !
1743 Calflag: ! check filament flags; if <.2 amps on filament, temporarily put .2 amps thru to check
1746 FOR I=0 TO 1   ! sample or preheat
1749   FOR J=1 TO 2   ! center or side
1752     IF Filament(I+2*J-1)<.2 THEN OUTPUT 8;Ff$(I,J),200
1755   NEXT J
1758 NEXT I
1761 OUTPUT KBD;Clear$;
1764 CALL Flag(Flag(*),Mu)
1767 FOR I=0 TO 1 ! restore fil-currents if necessary
1770   FOR J=1 TO 2
1773     IF Filament(I+2*J-1)<.2 THEN OUTPUT 8;Ff$(I,J),FNF(Filament(I+2*J-1))
1776   NEXT J
1779 NEXT I
1782 GOTO Bmc
1785 !
1788 Collmenu:! menu for accessing collector functions
1791 GOSUB Clearall
1794 REDIM Response$(10)
1797 MAT Response$= ("")
1800 IF Mu THEN Response$(1)="USE FARADAY CUP AS COLLECTOR"
1803 IF Daly AND NOT Mu THEN Response$(1)="USE DALY-DETECTOR AS COLLECTOR"
1806 Response$(2)="TAKE ZEROES FOR BOTH COLLECTORS"
1809 Response$(3)="CALIBRATE DALY GAIN"
1812 Response$(4)="CHANGE ENABLE/DISABLE STATUS OF DALY"
1815 Response$(10)="RETURN TO BMC"
1818 CALL Keymenu(Response$(*))
1821 IF Mu THEN ON KEY 0 LABEL "0: CUP" GOTO 1845
1824 IF NOT Mu THEN ON KEY 0 LABEL "0: DALY" GOTO 1845
1827 ON KEY 1 LABEL "1:  ZEROES" GOTO Calzer
1830 ON KEY 2 LABEL "2:  DALYCAL" GOTO Dalycal
1833 ON KEY 3 LABEL "3:  DALYSTATUS" GOTO Dalystat
1836 ON KEY 9 LABEL "9:  ESCAPE" GOTO Skey_bmc
1839 GOTO 1839
1842!
1845 GOSUB Collchange
1848 GOTO Skey_bmc
1851 !
1854 Calzer:OUTPUT KBD;Clear$;  ! determine zeroes for Faraday Cup & Daly
1857 CALL Zero(Filament(*),0,0,Daly,B,Peak_inter,Mu,Mm$(*),Collector$(*))    ! determine zeroes for Faraday Cup & Daly
1860 IF Subflag THEN
1863   DISP "(press "&FNH$("CONTINUE")&" to continue)"
1866   PAUSE
1869 END IF
1872 GOTO Bmc
1875 !
1878 Dalycal:Center_peak(1,L)      !        Calibrate Daly gain
1881 IF Subflag=4 THEN Skey_bmc
1884 OUTPUT 8;Mm$(0,1)
1887 IF Mu THEN WAIT 6
1890 Calibrate(Filament(*),Magnet(L,2),Peak_inter,Daly,I_t,Mu,Mm$(*),Collector$(*))
1893 GOTO Bmc_1
1896 !
1899 Dalystat:GOSUB Daly
1902 GOTO Skey_bmc
1905 !
1908 Chartmenu: ! Menu for accessing graphics-stripchart functions
1911 GOSUB Clearall
1914 REDIM Response$(10)
1917 MAT Response$= ("")
```

```
2097 CASE -162,-169,-170,-172,-418,-426,-425,-428
2100   !change fil-current by .01 amps: unshifted U/D arrow for sample-center, shifted U/D arrow for sample-side, ditto but .001 am
ps if CTRL used
2103 Mfil:K=Keycode
2106   F=(K=-162 OR K=-170 OR K=-418 OR K=-426)+2*(K=-169 OR K=-172 OR K=-425 OR K=-428)
2109   IF F=2 AND Nfils=1 THEN Bmc_1
2112   Filnum=F
2115   SELECT K
2118   CASE -162,-169     ! up-arrow, SHIFT up-arrow
2121     Change=.01
2124   CASE -170,-172     ! down-arrow, SHIFT down-arow
2127     Change=-.01
2130   CASE -418,-425     ! CTRL up-arrow, CTRL-SHIFT up-arrow
2133     Change=.001
2136   CASE -426,-428     ! CTRL down-arrow, CTRL-SHIFT down-arrow
2139     Change=-.001
2142   END SELECT
2145   F=PROUND(Filament(Filnum)+Change,-4)
2148   Filament(Filnum)=(F)0)*(F*(F(=7)+7*(F)7))
2151   OUTPUT 8 USING "4A,4Z";Ff$((Filnum)2),1+(Filnum=2 OR Filnum=4)),FNF(Filament(Filnum))
2154   BEEP 220*(Change(0)+660*(Change)0),.05-.03*(Change(.01)
2157   MAT Daly_ok= (0)   ! zero the multiplier-permitted elements for all isotopes
2160   Daly_ok(L)=Mu   !    but restore the original value for present isotope
2163   GOTO Bmc_1
2166 END SELECT
2169 !
2172 OFF KEY
2175 SELECT Keycode
2178 CASE 24 !                    (CTL X)
2181   ALPHA OFF
2184   CALL Pattern
2187 CASE 17,18,19,20    !        (CTL 1,2,3, or 4)
2190   GOSUB Filoff
2193 CASE 15    !                  (CTL /)
2196   I_t=1  !                    1-second integration time
2199   PRINT TABXY(1,18);" *** 1 SECOND INTEGRATION TIME ***    (press CTL ^ to return to 0.2 seconds)"
2202   BEEP 200,.1
2205   GOTO Bmc
2208 CASE 30    !                   (CTL ^)
2211   I_t=2   !                   0.2-second integration time
2214   PRINT TABXY(1,18);"** 0.2 second integration time **"&RPT$(" ",47)
2217   BEEP 800,.1
2220   GOTO Bmc
2223 CASE 4    !                   (CTL-D)
2226   GOTO Dalystat
2229 CASE 63,47    !               (? or /)
2232   Shiftlabel(Stripchart,Sample_name$(*),Sample,Run,Full_auto,Nfils)
2235 CASE -179    !                (ALPHA key)
2238   OUTPUT KBD;CHR$(255)&CHR$(-179);
2241   Stripchart=0
2244 CASE -180    !                (GRAPHICS key
2247   OUTPUT KBD;CHR$(255)&CHR$(-180);
2250 CASE 8    !                   (CTL-H)
2253   GOSUB Querytime    !        Read clock
2256 CASE 7     !                  (CTL-G)
2259   GOTO Calpgraph    !         pressure-graphics
2262 CASE 22    !                  (CTL-U)
2265   GOTO Calhv    !             query HV
2268 CASE 6    !                   (CTL-F)
2271   StripchartO=Stripchart
```

```
1920 Response$(1)="START "&FNUn$("LINEAR")&" GRAPHICS BEAM-MONITOR"
1923 Response$(2)="START "&FNUn$("LOGARITHMIC")&" GRAPHICS BEAM-MONITOR"
1926 IF Candump AND NOT Dump_stripchart THEN Response$(5)="ENABLE AUTO PRTR-DUMP OF GRAPHICS BEAM-MONITOR"
1929 IF Candump AND Dump_stripchart THEN Response$(5)="DISABLE AUTO PRTR-DUMP OF GRAPHICS BEAM-MONITOR"
1932 Response$(3)="DOUBLE X-AXIS TIME-PERIOD (now "&VAL$(Xtime)&" mins)"
1935 Response$(8)="HALVE X-AXIS TIME-PERIOD (now "&VAL$(Xtime)&" mins)"
1938 Response$(4)="DOUBLE Y-AXIS HT (now "&VAL$(Yheight)&"X initial beam)"
1941 Response$(9)="HALVE Y-AXIS HT (now "&VAL$(Yheight)&"X initial beam)"
1944 Response$(10)="RETURN TO BMC"
1947 CALL Keymenu(Response$(*))
1950 ON KEY 0 LABEL "START LIN-GBM" GOTO Linmonitor
1953 ON KEY 1 LABEL "START LOG-GBM" GOTO Logmonitor
1956 IF Candump AND NOT Dump_stripchart THEN ON KEY 4 LABEL " ON AUTO-DUMP" GOTO Autdump
1959 IF Candump AND Dump_stripchart THEN ON KEY 4 LABEL "OFF AUTO-DUMP" GOTO Offdump
1962 ON KEY 2 LABEL "DOUBLE X-AXIS" GOTO Double_x
1965 ON KEY 7 LABEL " HALVE X-AXIS" GOTO Halve_x
1968 ON KEY 3 LABEL "DOUBLE Y-AXIS" GOTO Double_y
1971 ON KEY 8 LABEL " HALVE Y-AXIS" GOTO Halve_y
1974 ON KEY 9 LABEL "    ESCAPE" GOTO Skey_bmc
1977 IF Dump_stripchart THEN DISP FNH$("AUTO PRINTER-DUMP")&FNB1$("ENABLED")
1980 IF NOT Dump_stripchart THEN DISP FNH$("AUTO PRINTER-DUMP DISABLED")
1983 GOTO 1983
1986!
1989 Linmonitor:Logmon=0
1992 GOTO Monitor
1995 Logmonitor:Logmon=1
1998 GOTO Monitor
2001 Autdump:Dump_stripchart=1
2004 BEEP 660,.1
2007 GOTO Chartmenu
2010 Offdump:Dump_stripchart=0
2013 BEEP 220,.1
2016 GOTO Chartmenu
2019 Double_x:Xtime=2*Xtime
2022 BEEP 660,.05
2025 GOTO Chartmenu
2028 Halve_x:IF Xtime)=2 THEN Xtime=Xtime/2
2031 BEEP 220,.05
2034 GOTO Chartmenu
2037 Double_y:Yheight=2*Yheight
2040 BEEP 660,.05
2043 GOTO Chartmenu
2046 Halve_y:Yheight=Yheight/2
2049 BEEP 220,.05
2052 GOTO Chartmenu
2055 !
2058 !
2061 Calrunvar:CALL Runvariables(Runvar(*),Run_name$(*),Runtype$(*),Sample_name$(*),Run_order(*),Run_iso(*),0)   ! manipulate run va
riables
2064 GOTO Skey_bmc
2067!
2070!
2073! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2076 Keybranch:CALL Kybrd(KBD$,Keycode)  ! branch from non-softkey keystroke
2079 OFF KBD
2082 OFF KNOB
2085 SELECT Keycode  ! branching to avoid blinking of key-display by avoiding OFF KEY
2088 CASE 32,94,40,41,43,45,-196,-194,-463 TO -455,-414 TO -406
2091   IF L THEN GOSUB Magswitch
2094   GOTO Bmc_1
```

```
2274   GOTO Calmfocus   !          manual focus
2277 CASE 3 !                     (CTL-C)
2280   Check_elements(Element$(*),0) ! recheck element-types on disk
2283 CASE -193    !               (RECALL)
2286   IF NOT Auto THEN Bmc_1
2289 Resume_auto:OFF KEY          !     resume automatic running after interrupt
2292   ON KEY 0 LABEL "     BMC" RECOVER Recover_bmc
2295   ON KEY 1 LABEL "   NEXT RUN" RECOVER Recover_quit
2298   OFF KNOB
2301   Full_auto=1
2304   OUTPUT KBD;Clear$;
2307   GRAPHICS OFF
2310   Nfils=1+(Runvar(Run,1)>1)
2313   Spike=Runvar(Run,26)
2316   IF Spike THEN
2319     Whichspike(Spike,1,Subflag)
2322     IF Subflag THEN Autospike_error
2325   END IF
2328   Daly=Runvar(Run,5)*(Daly<>0)
2331   ON 1+Where GOTO Autcont,Autfind,Preheat,Filup,Takeup_cf,Tuneup_cfonly,Curr1,Curr_1a,Wait1,Curr2,Wait2,Start,Retake,Re1,Sf1,Dw
ait,Go,Take_data
2334   !
2337 CASE -192  !                  (SHIFT-RECALL)
2340   IF NOT Auto THEN Bmc_1
2343   INPUT "WHICH RUN DO YOU WANT TO CONTINUE AUTOMATIC-RUNNING FROM?  (0 to escape)",A
2346   IF A=0 THEN Bmc
2349   IF Runvar(A,1)=0 THEN
2352     PRINT "NO RUN VARIABLES DEFINED FOR RUN#";A
2355     Clunk
2358     GOTO 2343
2361   END IF
2364   Run=A
2367   Full_auto=1
2370   GOTO Autcont
2373 CASE 1  !                    (CTL-A)
2376   Average(0,Normal(*),Candump,E,(Run),Prtr(*))    ! calculate weighted averages
2379   IF E THEN Skey_bmc
2382 CASE 2  !                    (CTL-B)
2385   GOTO Caltest !             graphic contact-test for all samples
2388 CASE 13     !                (CTL -)
2391   IF Magnet(0,1) AND Magnet(0,2) AND L THEN
2394     From_iso=L  !             switch magnet to Re-187
2397     L=0
2400     Mu=0
2403     Correct(Ff$(*),Mm$(*),Filament(*),Magnet(0,2),Mu,I_t,Coarsemag(0),Foc(*))
2406     B=Magnet(L,2)
2409   END IF
2412 CASE 11     !                (CTL +)
2415   L=From_iso  !             switch magnet from 187 to normal peak
2418   Mu=0
2421   Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(1),Foc(*))
2424   B=Magnet(L,2)
2427 CASE 16    !                 (CTL-P)
2430   GOTO Calshape !             do a graphics check of peakshape
2433 CASE 14    !                 (CTL-N)
2436   Show_names(Sample_name$(*)) !   display all defined sample names
2439 CASE 12    !                 (CTL-L)
2442   GOTO Catalog   !            locate run-data on disk
2445 CASE 26    !                 (CTL-Z)
2448   GOTO Calzer
```

```
2451 CASE 10  !                    (CTL-*)
2454   Stripchart=0      !           do a complete center/focus/barrel/focus
2457   Center_peak(1,L)
2460   IF Subflag=4 THEN Skey_bmc
2463   Focus(3,1,1,Focnum)
2466   IF Subflag=4 THEN Skey_bmc
2469   Center_peak(1,L)
2472   Center_barrel(Sample)
2475   IF Subflag=1 THEN 807
2478   IF Subflag=4 THEN Skey_bmc
2481   Focus(2,1,1,Focnum)
2484 CASE 61      !                equals sign; calc. 207/206 age
2487   Mu=0
2490   OUTPUT 8;Mm$(Mu,I_t)   ! put on cup just in case
2493   A=0
2496   DISP FNH$("ENTER RADIOGENIC 207/206 RATIO")&"   (CONTINUE to escape)";
2499   INPUT A
2502   IF A THEN CALL Pb_age(A)
2505 CASE -177    !                (DUMP ALPHA)
2508   DUMP ALPHA
2511 CASE -178    !                (DUMP GRAPHICS)
2514   IF Candump THEN DUMP GRAPHICS
2517 CASE -195      !              (RESULT)
2520   P=2          !                printout isotope-ratio data for run(s)
2523   GOSUB Result       !          using printer (RESULT) or CRT (SH-RESULT)
2526   GOTO Skey_bmc
2529 CASE -163    !                (SHIFT-RESULT)
2532   P=1        !                 display isotope-ratio data for run(s)
2535   GOSUB Result
2538 CASE -176,-223               !(PAUSE, STOP)
2541   PAUSE   !                    pause the program
2544 CASE -181,-221               !(CLR LN, CLR SCR)
2547   OUTPUT KBD;Clear$; !          clear screen
2550 CASE -220                     ! SHIFT-STEP
2553   OUTPUT KBD;Clear$;
2556   PRINT USING "10/,K,/,K";"PRESS  k0  TO COMPLETELY RESTART THE PROGRAM","( this will turn off all filaments & reset the barr
el!! )"
2559   PRINT USING "2/,K";"PRESS  k9  TO RETURN TO BMC"
2562   OFF KEY
2565   ON KEY 0 LABEL "   RESTART" GOTO 2574
2568   ON KEY 9 LABEL "    ESCAPE" GOTO Skey_bmc
2571   GOTO 2571
2574   Debug=0
2577   GOTO 3
2580 CASE -214 !             INS LN key
2583   OUTPUT KBD;Clear$; !   dump graphics with a label at the bottom
2586   Input$=""
2589   LINPUT "GRAPHICS-DUMP WITH LABEL AT BOTTOM---   ENTER LABEL  ( ? to escape):",Input$[1,60]
2592   IF Input$[1,1]="?" THEN Skey_bmc
2595   CSIZE 4
2598   VIEWPORT 0,100*RATIO,0,100
2601   WINDOW 0,100,0,100
2604   LORG 1
2607   MOVE 5,15
2610   LABEL Input$[1,52]
2613   DUMP GRAPHICS
2616   GCLEAR
2619   Stripchart=0
2622   GOTO Skey_bmc
2625 CASE ELSE
```

```
2805 GOTO Bmc_1
2808 !
2811 !
2814 ! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2817 !
2820 Magswitch:  !  switch magnet to another position or isotope
2823 Key$=CHR$(Keycode)
2826 K=Keycode
2829 SELECT K
2832 CASE -463 TO -455 ! CTRL-softkey pressed; jump up # of masses of softkey
2835   Jump=K+464
2838   L=L+Jump
2841   IF L>Niso THEN L=Niso
2844   B=FNIsomag(Mcoef(*),(Magnet(L,1)))
2847   Beep(2000,.03,.02,5)
2850   Jump_dir=1
2853   J0=3
2856 CASE -414 TO -406 ! CTRL-SHIFT softkey pressed; jump down #of pks of skey
2859   Jump=-(K+415)
2862   L=L+Jump
2865   IF L<1 THEN L=1
2868   B=FNIsomag(Mcoef(*),(Magnet(L,1)))
2871   Beep(220,.03,.02,5)
2874   Jump_dir=-1
2877   J0=3
2880 CASE -196       ! left-arrow; switch magnet 1/2 mass down
2883   B=FNIsomag(Mcoef(*),Magnet(L,1)-.5)
2886   J0=1
2889   Beep(440,.03,.02,3)
2892 CASE -194       ! right-arrow;  switch magnet 1/2 mass up
2895   B=FNIsomag(Mcoef(*),Magnet(L,1)+.5)
2898   J0=5
2901   Beep(1000,.03,.02,3)
2904 CASE 32,94      ! space-bar or ^; switch to peak-top
2907   J0=3
2910   B=Magnet(L,2)
2913   Beep(660,.03,.02,3)
2916 CASE 40         ! left-parenthesis key; switch magnet to lower-side of peak
2919   B=Magnet(L,2)-Aside
2922   J0=2
2925   Beep(550,.03,.02,3)
2928 CASE 41         ! right-parenthesis key; switch magnet to upper-side of pk
2931   B=Magnet(L,2)+Aside
2934   J0=4
2937   Beep(880,.03,.02,3)
2940 CASE 43         ! + key; switch magnet up to next-defined isotope
2943   Jump_dir=1
2946   L=L+(L(Niso)
2949   B=Magnet(L,2)
2952   J0=3
2955   BEEP 660,.05
2958 CASE 45         ! - key; switch magnet down to next-defined isotope
2961   Jump_dir=-1
2964   L=L-(L>1)
2967   B=Magnet(L,2)
2970   J0=3
2973   BEEP 220,.05
2976 END SELECT
2979 Mu=Mu*Daly_ok(L)
2982 OUTPUT 8 USING Mag;B
```

```
2628   DISP CHR$(Keycode);"     (code ";Keycode;")   ??  - UNDEFINED KEY"   ! undefined key pressed during BMC
2631   Clunk
2634   WAIT .5
2637 END SELECT
2640 GOTO Bmc_1
2643   !
2646 Result:OFF KEY
2649 IF NOT Full_auto THEN
2652   OUTPUT KBD;Clear$;
2655   PRINT USING "12/,K,2/,K";"STARTING [, ENDING] RUNS FOR PRINTOUT? (CONTINUE = current run ["&VAL$(Run)&"], 0=escape)","(or u
se negative #s for file #s)"
2658   LINPUT Input$
2661   Parse(Input$,Val(*),Ninputs)
2664   IF Ninputs=0 THEN
2667     A=Run
2670     Z=Run
2673   ELSE
2676     A=Val(1)
2679     Z=Val(1+(Ninputs>1))
2682   END IF
2685   IF A=0 THEN RETURN
2688   E=0
2691   OUTPUT KBD;Clear$;
2694   IF A<0 AND Z<0 THEN  ! request file #s, not run #s
2697     Printres(Prtr(P),0,E,ABS(A),ABS(Z))
2700     IF E THEN RETURN
2703     RETURN
2706   END IF
2709 ELSE
2712   A=1
2715   Z=Run
2718   IF Full_auto THEN P=2
2721 END IF
2724 FOR Run_num=A TO Z
2727   IF NOT Auto OR NOT POS(UPC$(Runtype$(Run)),"OUTGAS") THEN CALL Printres(Prtr(P),(Run_num),E)
2730   IF E THEN RETURN
2733 NEXT Run_num
2736 RETURN
2739        !
2742 Monitor:IF NOT Logmon THEN  ! draw axes & start graphics beam-monitor
2745   IF Mu*(Mu<.1) OR (NOT Mu)*(Mu<2) THEN
2748     Ymax=Mu/10+(NOT Mu)*2        ! small beam
2751   ELSE
2754     Ymax=Mu*Yheight       ! not-small beam
2757     IF Ymax>10000 THEN Ymax=10000
2760   END IF
2763 ELSE
2766   IF Mu<=0 THEN Mu=.01
2769   Ymax=INT(LGT(Mu*Yheight))+3   ! logarithmic monitor
2772   IF Ymax>4 THEN Ymax=4
2775   IF Ymax<1 THEN Ymax=1
2778 END IF
2781 Stripchart=1
2784 Running_time=TIMEDATE
2787 Tmin=(Running_time-Time0)/60
2790 Running_time=Tmin
2793 Tmax=INT(Tmin)+Xtime
2796 Axes(0,100,25,100,INT(Tmin),Tmax,-Logmon*(1+2*Mu),Ymax,"MINUTES SINCE START OF RUN","mV BEAM",0,Mu,(Logmon))
2799 MOVE Tmin,Mu*(NOT Logmon)+LGT(Mu*(Mu>0)+(Mu<=0)*.00001)*Logmon
2802 OUTPUT KBD;Clear$;
```

```
2985 PRINT TABXY(1,18);RPT$(" ",80)
2988 RETURN
2991  !
2994  !
2997 Filoff:  !  turn a filament off
3000 Fnum=Keycode-16  ! =1,2,3, or 4 for filament#s 1,2,3,4
3003 Preheat=(Keycode>18)
3006 ON DELAY .5 GOTO 3066   ! refuse to comply if keystroke not repeated within .5 seconds
3009  !
3012 ON KBD GOTO 3027
3015 BEEP 220,.03
3018 WAIT .01
3021 GOTO 3015
3024  !
3027 Kybrd(KBD$,Keycode)
3030 IF (Keycode<17 OR Keycode>20) THEN 3066
3033  !
3036 OFF DELAY
3039 OFF KBD
3042 OUTPUT 8 USING "4A,4Z";Ff$(Preheat,1+(Fnum=2 OR Fnum=4)),0
3045 IF NOT Preheat THEN PRINT "SAMPLE ";
3048 IF Preheat THEN PRINT "PREHEAT ";
3051 PRINT Fil$(Fnum-2*Preheat);" OFF"
3054 Filament(Fnum)=0
3057 Mu=0
3060 GOTO Bmc_1
3063 !
3066 PRINT USING "5/,K,2/";"TO TURN OFF A FILAMENT, PRESS THE KEY TWICE DURING THE CHIRP."
3069 OFF DELAY
3072 OFF KBD
3075 RETURN
3078 !
3081 ! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3084 Mandat:GOSUB Clearall  ! Take data "manually" (e.g. by keyboard-request)
3087 IF Auto THEN
3090   PRINT USING "8/,K,2/,K";"YOU MUST EXIT THE AUTOMATIC-RUNNING MODE","IF YOU WANT TO TAKE DATA IN THE MANUAL MODE.  (press SHIF
T-k8 during BMC)."
3093   Clunk
3096   WAIT 5
3099   GOTO Skey_bmc
3102 END IF
3105 DATA 0,0,0,0,0
3108 RESTORE 3105
3111 READ Re,Auto,Full_auto,Beam_window,Limit_growth
3114 OFF KEY
3117 IF Run=0 THEN Run=1
3120 Start_time=Time0
3123 IF Run_name$(Run)<>"" AND NOT Changed_el THEN Manualdata_menu
3126!
3129! first block(s) of data requested for this run
3132 IF NOT Changed_el THEN
3135   MAT Lastblock_iso= (0)
3138   MAT Peak= (0)
3141   MAT Last_aver= (0)! no previous blocks of data
3144   Last_isotopes$=""
3147 END IF
3150 Dump_datagraf=0
3153  !
3156 IF Normal0(1) AND Normal0(2) THEN
3159   Fract_normal=1  ! normalize for fractionation using a natural ratio
```

```
3162 ELSE
3165   Fract_normal=0
3168 END IF
3171 IF NOT Fract_normal AND TRIM$(UPC$(Runclass$))<>"U" THEN Manualdata_menu
3174 ! include U element in case of 233-236 double-spiked run
3177    !
3180    ! more info needed if a normalizable element
3183 DATA  Sample Name,"",Spike Number (0 if an unspiked run),"",Normalize data to ratios of,"  First Block? (Y/N)",??,"","0","","",
NO
3186 DATA -1,-1,0,0,-1,-1,0,0,0,0,-2,-2
3189 RESTORE 3183
3192 REDIM Prompt$(6),Range(6,2),Use(6),Response$(6)
3195 READ Prompt$(*),Response$(*),Range(*)
3198 MAT Use= (1)
3201 Use(4)=0
3204 Use(2)=0
3207 Use(5)=0
3210 IF Sample_name$(Sample)<>"" THEN Response$(1)=Sample_name$(Sample)
3213 Form(Prompt$(*),Response$(*),Use(*),Range(*),6,1,E,2)
3216 IF E THEN Skey_bmc
3219!Temp$=Response$(1)  ???
3222 Sample_name$(Sample)=Response$(1)
3225 IF NUM(Response$(3))>47 AND NUM(Response$(3))<58 THEN
3228   Spike=VAL(Response$(3))
3231 ELSE
3234   DISP FNH$("THE SPIKE NUMBER MUST BE A DEFINED SPIKE WITH A NUMBER BETWEEN 1 AND 20")
3237   Clunk
3240   WAIT 4
3243   GOTO 3189
3246 END IF
3249 IF Spike THEN
3252   Whichspike(Spike,1,Subflag)
3255   IF Subflag THEN
3258     DISP FNH$("*** CAN'T GET SPIKE# "&VAL$(Spike)&" FROM EITHER DISK ****")
3261     Clunk
3264     WAIT 2
3267     GOTO Mandat
3270   END IF
3273 END IF
3276 Norm1=FNYes(Response$(6))
3279 IF Norm1 THEN Normal(1)=0
3282 !
3285 !
3288 DATA Sample Name,Isotopes,Number of Sets in a block,Number of Blocks,Beam Window (most-intense peak):,"  Minimum Beam (volts)",
"  Maximum Beam (volts)"
3291 DATA "  Maximum Filament-Current (amps)",Maximum Beam-Growth (%/minute),Final Filament-Currents (amps),"  (sa-Cen,sa-Si,preh-Ce
n,preh-Si)"
3294 DATA"Daly Status (0,1,2)","   {0: Disabled    1: OK for data","    2: Beam-tuneup only}",Dump Graphics each Block? (Y/N)
3297 DATA ??,??,"15","1","","0","10","6","100","","","","","","NO"
3300 DATA -1,-1,0,0,4,40,1,80,0,0,0,10,0,10,0,6,0,100,0,0,0,0,0,0,0,0,0,2,-2,-2
3303!
3306 Manualdata_menu:RESTORE 3288  ! Construct Data-Taking Form
3309 REDIM Prompt$(15),Response$(15),Use(15),Range(15,2)
3312 READ Prompt$(*),Response$(*),Range(*)
3315 IF Sample_name$(Sample)<>"" THEN Response$(1)=Sample_name$(Sample)
3318 MAT Use= (1)
3321 DATA 0,0,0,0
3324 READ Use(5),Use(10),Use(12),Use(13)
3327 IF Daly THEN  ! only allow Daly-enable if Daly not completely disabled
3330   Response$(14)=VAL$(Daly)
```

```
3333 ELSE
3336   Use(14)=0
3339   Prompt$(12)=""
3342   Prompt$(13)=""
3345   Prompt$(14)=""
3348   Response$(14)=""
3351 END IF
3354 IF Candump=0 THEN
3357   Use(15)=0
3360   Prompt$(15)=""
3363   Response$(15)=""
3366 ELSE
3369   Response$(15)=CHR$(89*Dump_datagraf+78*(NOT Dump_datagraf))
3372 END IF
3375  !
3378 IF Normal0(1)=0 OR Normal0(2)=0 THEN Fract_normal=0  !shouldn't be necessary, but sometimes an invalid fract_normal of 1 has ap
peared anyway
3381 IF Fract_normal OR TRIM$(UPC$(Runclass$))="U" AND Block=0 THEN Response$(1)=Sample_name$(Sample)
3384 IF NOT Fract_normal AND TRIM$(UPC$(Runclass$))<>"U" THEN
3387   IF Block=0 AND Sample_name$(Sample)<>"" THEN Response$(1)=Sample_name$(Sample)
3390   IF Block<>0 AND Sample_name$(Sample)="" THEN Response$(1)="??"
3393 END IF
3396 IF Block THEN
3399   Response$(1)=Run_name$(Run)
3402   Response$(2)=Last_isotopes$
3405   Response$(3)=VAL$(Nsets)
3408   Response$(4)=VAL$(Numberofblocks)
3411   Response$(6)=VAL$(Wmin)
3414   Response$(7)=VAL$(Wmax)
3417   Response$(8)=VAL$(Fmax)
3420   Response$(9)=VAL$(Maxgrowth)
3423   IF Candump THEN Response$(15)=CHR$(89*Dump_datagraf+78*(NOT Dump_datagraf))
3426 END IF
3429 !
3432 IF Spike THEN
3435   Prompt$(2)=Prompt$(2)&" (include "&VAL$(Spkdrun_ref)&", "&VAL$(Spikedrun_iso(1))&", & "&VAL$(Spikedrun_iso(2))&")"
3438   IF Block=0 THEN
3441     Response$(2)=VAL$(Spkdrun_ref)&","&VAL$(Spikedrun_iso(1))&","&VAL$(Spikedrun_iso(2))
3444     IF Spikedrun_iso(3) THEN Response$(2)=Response$(2)&","&VAL$(Spikedrun_iso(3))
3447   END IF
3450 END IF
3453  !
3456 IF NOT Spike AND NOT Fract_normal THEN Prompt$(2)=Prompt$(2)&" (Reference-Isotope First)"
3459 IF NOT Spike AND Fract_normal THEN
3462   Prompt$(2)=Prompt$(2)&" (must include "&VAL$(Ref)&" and "&VAL$(Normal0(1))&")"
3465   IF NOT Block THEN Response$(2)=VAL$(Ref)&","&VAL$(Normal0(1))
3468 END IF
3471 IF Block=0 AND Spike=0 THEN
3474   IF UPC$(Runclass$[1,2])="PB" THEN Response$(2)="206,207,208,204"
3477   IF UPC$(Runclass$[1,2])="RB" THEN Response$(2)="87,85"
3480   IF UPC$(Runclass$[1,2])="TH" THEN Response$(2)="232,230"
3483   IF UPC$(Runclass$[1,1])="U" THEN Response$(2)="238,235"
3486   IF UPC$(Runclass$[1,3])="UO2" THEN Response$(2)="270,267"
3489   IF UPC$(Runclass$[1,2])="SR" THEN Response$(2)="86,88,87"
3492   IF UPC$(Runclass$[1,2])="ND" THEN Response$(2)="144,146,143"
3495   IF UPC$(Runclass$[1,3])="NDO" THEN Response$(2)="160,162,159"
3498   IF UPC$(Runclass$[1,1])="K" THEN Response$(2)="39,41"
3501 END IF
3504!
3507 Response$(11)=VAL$(Filament(1))&","&VAL$(Filament(2))&","&VAL$(Filament(3))&","&VAL$(Filament(4))
```

```
3510 Invoke_menu:Form(Prompt$(*),Response$(*),Use(*),Range(*),15,1+(Response$(1)()"??"),E,1)
3513 IF E THEN Skey_bmc
3516 OFF KEY
3519 Run_name$(Run)=Response$(1)
3522 Sample_name$(Sample)=Response$(1)
3525 Nsets=VAL(Response$(3))
3528 Numberofblocks=VAL(Response$(4))
3531 Wmin=VAL(Response$(6))
3534 Wmax=VAL(Response$(7))
3537 IF Wmax<1.2*Wmin THEN  ! beam-window must be at least 20% wide
3540   DISP FNH$("MAX. BEAM MUST BE AT LEAST 20% GREATER THAN MIN. BEAM")
3543   Clunk
3546   WAIT 3
3549   GOTO Invoke_menu
3552 END IF
3555 Beam_window=(Wmin<>0 OR Wmax<>10)
3558 IF Daly THEN Daly=VAL(Response$(14))
3561 Fmax=VAL(Response$(8))
3564 Maxgrowth=VAL(Response$(9))
3567 Limit_growth=(Maxgrowth<>100)
3570 Temp$=Response$(11)
3573 Parse(Temp$,Val(*),Ninputs)
3576 MAT Off_fil= Filament
3579 Change_curr=0  ! change fil-currs after block?
3582 FOR I=1 TO 4
3585   Off_fil(I)=Val(I)
3588   Change_curr=Change_curr+ABS(Filament(I)-Off_fil(I))
3591 NEXT I
3594 IF Change_curr<.001 THEN Change_curr=0
3597 Dump_datagraf=Candump*FNYes(Response$(15))
3600 IF NOT Candump THEN Response$(15)=""
3603 !
3606 FOR I=1 TO 8
3609   Run_iso(Run,I)=0
3612 NEXT I
3615 Parse((Response$(2)),Val(*),Ninputs)
3618 IF Ninputs<2 THEN
3621   Clunk
3624   PRINT TABXY(1,12);Ci$;" **** YOUR "&Q$&"ISOTOPES"&Q$&" RESPONSE OF "&Response$(2)&" DOESN'T GIVE ME AT LEAST 2 ISOTOPES ****
";Cn$
3627   GOTO 3654
3630 END IF
3633 MAT Iso= (0)
3636 FOR I=1 TO Ninputs
3639   FOR J=1 TO Niso
3642     IF Val(I)=Magnet(J,1) THEN 3666
3645   NEXT J
3648   DISP Ci$;" **** ";Val(I);"ISN'T ON THE LIST OF DEFINED ISOTOPES FOR THIS ELEMENT-SERIES **** ";Cn$
3651   Clunk
3654   WAIT 3
3657   Response$(2)="??"
3660   GOTO Invoke_menu
3663    !
3666   Run_iso(Run,I)=Val(I)
3669   Iso(I)=Val(I)
3672 NEXT I
3675 IF Spike THEN    ! check compatibility of ELEMENT and SPIKE
3678   Spike0=Spike
3681   CALL Spike_check(Spike0,Iso(*),Spikedrun_iso(*),Spkdrun_ref)  ! check if the isotopes required by the spike are present
3684   IF Spike0=0 THEN
```

```
3687     Spike=0
3690     DISP FNH$("**** THE ISOTOPES FOR SPIKE# "&VAL$(Spike)&"DON'T MATCH ELEMENT "&Runclass$&" ***")
3693     Clunk
3696     WAIT 2
3699     GOTO Mandat
3702   END IF
3705 END IF
3708 IF Spike THEN Fract_normal=0
3711 IF Fract_normal THEN
3714   IF Val(1)<>Ref THEN  ! 1st isotope must be the reference-isotope for fractionation-normalized runs
3717     DISP FNH$(" THE FIRST ISOTOPE")&FNB1$("MUST")&FNH$("BE")&Ci$&VAL$(Ref)&" "&Cn$
3720     GOTO 3651
3723     IF NOT Spike THEN
3726       FOR I=2 TO Ninputs
3729         IF Val(I)=Normal(1) THEN 3753
3732       NEXT I
3735       DISP FNH$(" YOU MUST INCLUDE")&FNB1$(VAL$(Normal(1)))&FNH$("AS ONE OF THE ISOTOPES")
3738       GOTO 3651
3741     END IF
3744   END IF
3747 END IF
3750 !
3753 Last_isotopes$=Response$(2)
3756 !
3759 Ref=Run_iso(Run,1)  ! "Reference" isotope (either numerator or denominator of all ratios)
3762 Roughscan_done=1
3765 FOR I=1 TO 8  ! match the present-block's isotopes against previous to see if need a new rough-scan
3768   IF (Run_iso(Run,I)<>Lastblock_iso(I)) THEN Roughscan_done=0
3771 NEXT I
3774 IF SUM(Peak)=0 THEN Roughscan_done=0
3777 IF Roughscan_done=0 THEN
3780   MAT Peak= (0)
3783   MAT Order= (0)
3786   MAT Lastblock_iso= (0)
3789   GOSUB Mstart
3792   S=0
3795   FOR I=1 TO 8
3798     S=S+Run_iso(Run,I)
3801   NEXT I
3804   Pb_4678=(S=825 AND Ref=206) ! if a Pb-206-207-208-204 block
3807 END IF
3810  !
3813 PRINTER IS Prtr(2)
3816 IF Change_curr OR Beam_window OR Limit_growth THEN PRINT
3819 IF Change_curr THEN PRINT "FILAMENT CURRENTS AFTER LAST BLOCK: (sa-C,sa-S,Preh-C,Preh-S)";Off_fil(1);Off_fil(2);Off_fil(3):Off_
fil(4);"AMPS."
3822 IF Beam_window THEN PRINT "BEAMSIZE TO BE RESTRICTED BETWEEN";Wmin;"&";Wmax;"V  (for fil-currents(";Fmax;"A)"
3825 IF Limit_growth THEN PRINT "MAXIMUM PERMITTED GROWTH-RATE OF BEAM IS";Maxgrowth;"%/MINUTE."
3828 PRINTER IS CRT
3831 Changed_el=0
3834  !
3837 Gronk:FOR Blocknumber=1 TO Numberofblocks
3840   IF UPC$(Runclass$[1,2])="HF" THEN
3843      !
3846      ! Hafnium runs only
3849     OUTPUT 8;Mm$(0,1)  ! restrict the Re-187 beam to (8V if a hafnium run
3852     WAIT 6*Mu
3855     L=0  ! Re-187
3858     Mu=0
3861     Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),0,1,Coarsemag(0),Foc(*))
```

```
3864    WAIT 1
3867    Size(5,8,0,6,6,1,100,1)  ! 5 to 8 volts 187
3870    IF Subflag>1 THEN Bmc
3873    L=Rf  ! ref-peak
3876    Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),0,1,Coarsemag(1),Foc(*))
3879    WAIT 1
3882  END IF
3885 !
3888  IF Blocknumber>1 THEN Roughscan_done=1
3891  GOSUB Mgo
3894  IF Norm1 AND (Block=1) THEN
3897    FOR I=1 TO N-1  !  normalize to 1st-block value
3900      IF POS(Ratio$(I),VAL$(Bnorm)) THEN
3903        Normal(1)=Bnorm
3906        Normal(2)=Aver(I)^Inv
3909      END IF
3912    NEXT I
3915  END IF
3918 NEXT Blocknumber
3921 !
3924 FOR K=1 TO 4    ! change filament-currents to specified after-run settings
3927  IF (K)2 OR Change_curr) AND Filament(K)()Off_fil(K) THEN CALL Filament(10,Off_fil(K),.5,(K))
3930 NEXT K
3933 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L()0),Foc(*))
3936 GOTO Bmc
3939 ! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3942 !
3945 Mfind:GOSUB Clearall  ! Request a new sample to be rotated into position
3948 IF Run THEN PRINT TABXY(1,18);"(press "&FNH$("CONTINUE")&" with no response to escape)"
3951 A=0
3954 LINPUT "ENTER THE BARREL# OF THE NEW SAMPLE, NUMBER OF FILAMENTS FOR THIS SAMPLE?",Input$
3957 Parse(Input$,Val(*),Ninputs)
3960 A=Val(1)
3963 IF A=0 THEN Skey_bmc
3966 IF A<0 OR A>16 THEN 3978
3969 Sample=A
3972 Nfils=(Val(2)=1)+2*(Val(2)=2 OR Val(2)=3)
3975 IF Nfils=0 THEN
3978   PRINT TABXY(1,12);" PLEASE ENTER  2  VALUES; THE BARREL#,  AND THE NUMBER OF FILAMENTS  (1 OR 3)
FOR THE SAMPLE. "
3981   Clunk
3984   GOTO 3948
3987 END IF
3990 OUTPUT KBD;Clear$;
3993 Find(0,Sample,Nfils,Subflag,Filament(*),Time0)
3996 Changed_el=0
3999 Lastfil=0
4002 IF Daly THEN Daly=1 ! unless Daly broken, don,t inherit last run's Daly setting
4005 IF Block THEN Run=1+Run    ! increment Run# if took any data for previous sample
4008 IF Run>32 OR Run=0 THEN Run=1    ! Max. Run# is 32
4011 Run_name$(Run)=""
4014 Filnum=1
4017 MAT Normal= Normal0
4020 Block=0
4023 Focnum=0
4026 Lastrate=10
4029 Spike=0
4032 GOSUB Stdfocus
4035 Whoop
4038 Zero(Filament(*),Zero_time,Noise_time,Daly,Magnet(L,2),Peak_inter,Mu,Mm$(*),Collector$(*))
```

```
4041 RETURN
4044 !
4047 Stdfocus:  ! Restore std-focus values for single or triple filament samples
4050 ON ERROR GOTO Bad_read
4053 ASSIGN @Path1 TO "FOCUS:INTERNAL"
4056 ENTER @Path1,Nfils;Foc(*)
4059 OFF ERROR
4062 PRINT "STANDARD FOCUS-SETTINGS RESTORED"
4065 PRINT USING "K,2/,8(3D,3X),/,8(4D,2X),/";"FOCUSING VALUES:",1,2,3,4,5,6,7,8,Foc(*)
4068 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L<>0),Foc(*))
4071 RETURN
4074  !
4077 Clearall:GOSUB Clearkey
4080 OUTPUT KBD;Clear$;            ! clear alpha screen
4083 OFF KBD
4086 GRAPHICS OFF
4089 Stripchart=0
4092 RETURN
4095  !
4098 Skey_bmc:Shiftlabel(Stripchart,Sample_name$(*),Sample,Run,Full_auto,Nfils)
4101 OUTPUT 8;Mm$(Mu,I_t)
4104 GOTO Bmc_1
4107 !
4110 Clearkey:!               Undefine all softkeys & return
4113 FOR Iz=0 TO 19
4116   ON KEY Iz CALL Clunk
4119 NEXT Iz
4122 RETURN
4125  !
4128 Bad_store:OFF ERROR   ! failed disk-storage operation
4131 PRINT USING "2/,K";FNH$("UNABLE TO STORE DATA ON DISK")
4134 Clunk
4137 WAIT 2
4140 GOTO Bmc_1
4143 Bad_read:OFF ERROR   ! failed disk-read operation
4146 PRINT USING "2/,K";FNH$("UNABLE TO READ DATA FROM DISK")
4149 Clunk
4152 WAIT 2
4155 RETURN
4158 !
4161 Recover_bmc:I_t=2 ! BMC key (k0) pressed during auto-running,
4164 PRINTER IS CRT    ! so revert to BMC condition
4167 Full_auto=0
4170 From_iso=Rf
4173 A=KNOBX
4176 OFF KEY
4179 Broop
4182 PRINTER IS Prtr(2)
4185 PRINT USING "/,K,/";"++++ Exited from auto-running (BMC-key),  "&FNClock_12$(TIME$(TIMEDATE))&" ++++"
4188 GOTO Skey_bmc
4191 !
4194 Recover_quit:OUTPUT KBD;Clear$;  ! QUIT key (k2) pressed during auto-run,
4197 GRAPHICS OFF                     ! so quit this auto-run & go on to next
4200 Broop
4203 PRINTER IS Prtr(2)
4206 PRINT USING "/,K,/";"++++ Exited from auto-running (QUIT-key),  "&FNClock_12$(TIME$(TIMEDATE))&" ++++"
4209 PRINTER IS CRT
4212 Whoop
4215 OFF KEY
4218 A=KNOBX
```

```
4221 PRINT TABXY(1,9);"PRESS  k1  TO CONFIRM REQUEST TO QUIT THIS RUN AND GO ON TO NEXT RUN,"
4224 PRINT TABXY(1,11);"PRESS  k9  TO DENY REQUEST AND RESUME AUTOMATIC RUNNING OF THIS SAMPLE."
4227 ON KEY 1 LABEL "  QUIT RUN" GOTO Quit
4230 ON KEY 9 LABEL "    ESCAPE" GOTO Resume_auto
4233 GOTO 4233
4236  !
4239 Auto:! ********************AUTOMATIC RUNNING SEGMENT**********************
4242 ! *************************************************************************
4245 GOSUB Clearkey
4248 OFF KNOB
4251 DISP
4254 IF Auto THEN
4257   PRINT USING "18/,K,/";"PRESS  k2  TO RESUME AUTOMATIC RUNNING WHERE YOU LEFT OFF,"
4260   PRINT USING "K,2/,K,2/,K";"PRESS  k0  TO COMPLETELY RE-START AUTOMATIC RUNNING,","PRESS  k4  TO REVERT TO MANUAL RUNNING,
","PRESS  k9  TO ESCAPE."
4263 ELSE
4266   PRINT USING "18/,40X,K,2/,40X,K";"PRESS  k4  TO START AUTOMATIC RUNNING,","PRESS  k9  TO ESCAPE."
4269 END IF
4272 GOSUB Clearkey
4275 ON KEY 9 LABEL "    ESCAPE" GOTO Skey_bmc
4278 IF Auto THEN
4281   ON KEY 2 LABEL " RESUME AUTO" GOTO Resume_auto
4284   ON KEY 0 LABEL "   NEW AUTO" GOTO 4323
4287   ON KEY 4 LABEL "    MANUAL" GOTO 4305
4290 ELSE
4293   ON KEY 4 LABEL "  START AUTO" GOTO 4323
4296 END IF
4299 GOTO 4299
4302 !
4305 Auto=0
4308 Full_auto=0
4311 Spike=0
4314 I_t=2
4317 GOTO Skey_bmc
4320 !
4323 DATA 1,1,1,0,1,0,0
4326 RESTORE 4323
4329 READ Auto,I_t,Run,Repeat_run,Full_auto,Dump_stripchart,Dump_datagraf
4332 MAT Preheated= (0)
4335 GOSUB Clearkey
4338 ON KEY 0 LABEL "   DEFINED" GOTO 4365
4341 ON KEY 2 LABEL "   NOT SURE" GOTO 4359
4344 ON KEY 4 LABEL "  UNDEFINED" GOTO 4359
4347 ON KEY 9 LABEL "    ESCAPE" GOTO 4305
4350 PRINT USING "18/,16X,K,8/";"ARE YOUR AUTOMATIC-RUN VARIABLES DEFINED?"
4353 GOTO 4353
4356!
4359 OFF KEY
4362 CALL Runvariables(Runvar(*),Run_name$(*),Runtype$(*),Sample_name$(*),Run_order(*),Run_iso(*),1)
4365 IF Runvar(1,1)=0 OR Runtype$(1)="" THEN
4368   PRINT USING "2/,K,2/";FNH$("THERE ARE NO RUN VARIABLES IN MEMORY- NEED TO DEFINE OR GET FROM DISK ")
4371   Clunk
4374   ON KEY 9 LABEL "   ESCAPE" GOTO 4305
4377   WAIT 3
4380   OFF KEY 9
4383   GOTO 4359
4386 END IF
4389 OUTPUT KBD;Clear$;
4392 !
4395 Just_outgassing=1 ! Is this auto-run sequence just for outgassing?  If so
```

```
4398 FOR I=1 TO 32     ! don't check the HV & don't wory about the beam-valve.
4401   IF Runvar(I,1) AND POS(UPC$(Runtype$(I)[1,8]),"OUTGAS")=0 THEN Just_outgassing=0
4404 NEXT I
4407 !
4410 IF NOT Just_outgassing THEN
4413   CALL Hv(Mm$(*),Hv,0,Mu,I_t)
4416   ON KEY 0 LABEL "    READY" GOTO Autcont
4419   ON KEY 4 LABEL "    RECHECK" GOTO 4365
4422   ON KEY 9 LABEL "    ESCAPE" GOTO 4305
4425   PRINT TABXY(1,14);"CHECK THAT THE HV IS OK FOR ALL OF YOUR RUNS, AND THAT THE BEAMVALVE IS "&FNH$("OPEN")
4428   GOTO 4428
4431 END IF
4434 !
4437 Autcont: ! Where variable indicates program line where auto-running should be resumed if interrupted
4440 OFF KEY
4443 OFF KBD
4446 Where=0
4449 PRINTER IS CRT
4452 Outgas=POS(UPC$(Runtype$(Run)),"OUTGAS") ! just outgassing for this run
4455 Pregas=Outgas AND (UPC$(Runtype$(Run)[1,1])="P") ! outgassing in preheat position
4458 IF NOT Pregas THEN
4461   Sample=Run_order(Run)  ! sample number (1-16)
4464 ELSE
4467   Sample=Sample-1+17*(Sample=1) ! to outgas in preheat position for barrel# N, rotate barrel# n-1 into running position
4470 END IF
4473 Nfils=1+(Runvar(Run,1)>1)  ! Nfils is 1 if a single-filament sample, 2 if a triple
4476 Type0=Type
4479 Goer=0    ! until have a satisfactory beam
4482 IF Outgas THEN Autfind
4485 !
4488 FOR Type=1 TO 20  ! find appropriate element from disk, assuming original disk in drive
4491   IF Element$(Type)<>"" AND TRIM$(UPC$(Element$(Type)))=UPC$(Runtype$(Run)) THEN 4545
4494 NEXT Type
4497 ON ERROR GOTO 4518
4500 ASSIGN @Path1 TO "TYPE:INTERNAL"
4503 FOR Type=1 TO 20  ! look at each element-file on disk (maybe changed?)
4506   ENTER @Path1,Type;Runclass$,Niso,Mcoef(*),Magnet(*),Coarsemag(*),Peak_inter,Aside,Rf,Nuclide$(*),Normal(*),Inv,Interfere(*),H
v0
4509   IF TRIM$(UPC$(Runclass$))=TRIM$(UPC$(Runtype$(Run))) THEN 4545
4512 NEXT Type
4515 !
4518 OFF ERROR
4521 FOR I=2 TO 1 STEP -1
4524   PRINTER IS Prtr(I)
4527   PRINT USING "8/,K,DD,K,8/";Cb$&" ELEMENT '"&Runtype$(Run)&"' NOT RECOGNIZED --  RUN#",Run,Cn$
4530 NEXT I
4533 Superclunk
4536 Wait(TIMEDATE,20,0,Magnet(L,1),Auto,Full_auto)
4539 GOTO Do_next_run
4542 !
4545 OFF ERROR
4548 IF (Type0<>Type) OR (Run=1) THEN GOSUB Autel  ! get element-series from disk if not already in memory
4551 FOR I=1 TO 2 ! check HV twice in case of dropouts
4554   CALL Hv(Mm$(*),Hv,P,Mu,I_t)
4557   IF P OR (ABS(Hv-Hv0)<20) THEN 4584
4560 NEXT I
4563 !
4566 PRINTER IS Prtr(2)
4569 PRINT USING "3/,K,4D,K,8/";Cb$&" CAN'T RUN  "&Element$(Type)&"  ELEMENT-SERIES AT HIGH-VOLTAGE OF ",Hv,Cn$
4572 Superclunk
```

```
4575 Wait(TIMEDATE,20,0,Magnet(L,1),Auto,Full_auto)
4578 GOTO Do_next_run
4581   !
4584 Spike=Runvar(Run,26)
4587 IF NOT Normal0(1) AND NOT Spike AND Ref<>Run_iso(Run,1) THEN Ref=Run_iso(Run,1)  ! accept arbitrary reference-isotope if possib
le
4590 IF (Repeat_run=0) AND (Type0<>Type) THEN GOSUB Stdfocus
4593 Daly=Runvar(Run,5)
4596 Rb_like=(Nfils=2 AND Runvar(Run,2)<>187 AND Runvar(Run,3) AND Runvar(Run,4)) ! treat as an "Rb" run if a triple with a non-Re f
ocussing isotope
4599 ! for an "Rb-like" run, the center-fil is taken up until a non-Re beam of specified intensity is found, then left at that curre
nt throughout the run
4602 IF Spike THEN
4605   Whichspike(Spike,1,Subflag)
4608   IF Subflag THEN
4611 Autospike_error:PRINTER IS Prtr(2)
4614     PRINT USING "2/,3(K),10/";Cb$&" CAN'T GET SPIKE# ";Spike;" FROM DISK.  ABORTING RUN ."&Cn$
4617     Superclunk
4620     WAIT 20
4623     GOTO Do_next_run
4626   END IF
4629 END IF
4632 !
4635 PRINTER IS CRT
4638    !
4641 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(Rf,2),0,1,Coarsemag(L<>0),Foc(*))
4644 Where=1
4647 Autfind:OUTPUT KBD;Clear$;
4650 OFF DELAY
4653 PRINTER IS Prtr(2)
4656 IF Run=1 THEN PRINT USING "6/,7X,5(K),6/";"--- AUTOMATIC RUNNING STARTED AT  ",FNClock_12$(TIME$(TIMEDATE)),",   ",DATE$(TIMEDA
TE)," ---"
4659 IF Run>1 AND NOT Outgas THEN PRINT USING "3/,10X,7(K),2/";"--- STARTED RUN# ",Run," AT  ",FNClock_12$(TIME$(TIMEDATE)),",   ",D
ATE$(TIMEDATE)," ---"
4662 PRINTER IS CRT
4665 IF NOT Repeat_run THEN
4668   FOR I=1 TO 2
4671     Find(0,Sample,Nfils,Subflag,Filament(*),Time0)
4674     IF Pregas OR Subflag=0 THEN Found_sample
4677     PRINT USING "K,X,0,X,K,/";"**** FAILED TO FIND SAMPLE ON TRY#",I
4680   NEXT I
4683   IF Subflag THEN
4686     FOR P=2 TO 1 STEP -1
4689       PRINTER IS Prtr(P)
4692       PRINT USING "4/,3(K),4/";Cb$&"******* COULDN'T GET CONTACTS FOR SAMPLE AT BARREL# ",Sample," *******"&Cn$
4695     NEXT P
4698     Superclunk
4701     Wait(TIMEDATE,20,0,Magnet(L,1),Auto,Full_auto)
4704     GOTO Abort
4707   END IF
4710 END IF
4713  !
4716  !
4719 Found_sample:Focnum=0
4722 Lastfil=0
4725 Badflag=0 ! no lost contacts yet
4728 Fail_checkflags=0
4731 IF NOT Outgas THEN
4734   Sample_name$(Sample)=Run_name$(Run)
4737   Zero(Filament(*),Zero_time,Noise_time,Daly,Magnet(Rf,2),Peak_inter,Mu,Mm$(*),Collector$(*))
```

```
4740   IF Subflag THEN  ! bad zeroes- can't run
4743     Superduperclunk
4746     GOTO Recover_bmc
4749   END IF
4752 END IF
4755 !
4758 Where=2
4761 Preheat:CALL Flag(Flag(*),Mu)   ! take up preheat filament-currents
4764 Preheat_sample=Sample+1-16*(Sample=16)
4767 F=3*((Runvar(Run,24)<>0) AND (Runvar(Run,25)=0))+4*(Runvar(Run,25)<>0)     ! triple or single?
4770 IF F=4 AND ((Flag(2)=1) OR (Flag(2)=3)) THEN  ! triple-filament
4773   CALL Filament(60-50*(Runvar(Run,24)<1.5),Runvar(Run,24),.5,3)
4776 END IF
4779 IF (F=3 AND Flag(2)=1) OR (F=4 AND Flag(2)>1) THEN  ! take up sample filament(s)
4782   V=21+F
4785   IF Runvar(Run,V)>1.5 THEN CALL Filament(30,.7,1,(F))
4788   R=Runvar(Run,V)-Filament(F) ! remaining amperes to be gained
4791   CALL Filament(10,Runvar(Run,V)-.7*R,.4,(F)) ! next 70% at rate 10
4794   CALL Filament(3,Runvar(Run,V)-.2*R,.4,(F))  ! next 20% at rate 3
4797   CALL Filament(1,Runvar(Run,V),.4,(F))       ! last 10% at rate 1
4800 END IF
4803 IF Filament(3)>.3 THEN Preheated(Preheat_sample,1)=Filament(3)
4806 IF Filament(4)>.3 THEN Preheated(Preheat_sample,2)=Filament(4)
4809 IF Filament(3)>.3 OR Filament(4)>.3 THEN ON DELAY 60*120 GOSUB Off_preheat ! turn of preheat-fils after 2 hours
4812  !
4815 Where=3
4818 Filup: ! bring filament-currents to running temperature
4821 MAT Normal= Normal0
4824 !
4827 GOSUB Key_escape
4830 GOTO 4848
4833 !
4836 Key_escape:ON KEY 0 LABEL "     BMC" RECOVER Recover_bmc
4839 ON KEY 1 LABEL "   NEXT RUN" RECOVER Recover_quit
4842 RETURN
4845 !
4848 IF Outgas THEN  ! just outgassing
4851   Block=0
4854   Fil1=1+2*Pregas ! filament #s for outgassing- center
4857   Fil2=2+2*Pregas !    "     "   "      "     - sides
4860   Fil=Fil1*(Runvar(Run,1)=1)+Fil2*(Runvar(Run,1)=3) ! sample filament (center or side)
4863   OUTPUT 8 USING "4A,4Z";Ff$(Pregas,2),200 ! put .2 amps thru side-fil for valid flag-check
4866   IF Fil=2 THEN CALL Filament(20,Runvar(Run,4),.25,(Fil1)) ! takeup cf of triple
4869   CALL Filament((Runvar(Run,7)),Runvar(Run,6),.25,(Fil)) ! takeup sample-fil
4872   Wait(TIMEDATE,Runvar(Run,8)*60,Filament(Fil),Magnet(L,1),Auto,Full_auto)
4875   CALL Filament((Runvar(Run,10)),Runvar(Run,9),.25,(Fil))
4878   Wait(TIMEDATE,Runvar(Run,11)*60,Filament(Fil),Magnet(L,1),Auto,Full_auto)
4881   PRINTER IS Prtr(2)
4884   PRINT "Sample at Barrel#";TAB(18);Sample;TAB(22);"Outgassed at";TAB(34);Filament(Fil1);TAB(41);"Amps (Center)";
4887   IF Filament(Fil2) THEN PRINT ", ";TAB(56);Filament(Fil2);TAB(63);"Amps (Side)";
4890   PRINT USING "4/"
4893   PRINTER IS CRT
4896   GOTO Do_next_run
4899 END IF
4902 !
4905 IF Spike THEN MAT Normal= (0)
4908 Block=0
4911 GOSUB Crtlabel
4914 Start_wait=TIMEDATE
4917 Start_time=TIMEDATE ! for assigning times to data-blocks
```

```
4920 IF Nfils=2 THEN  ! triple-filament run
4923   ! Center-filament take-up for triple-filament runs
4926   IF Runvar(Run,2)=0 THEN L=0
4929   IF Runvar(Run,2) THEN
4932     FOR L=0 TO Niso  ! find focussing-peak index#
4935       IF Magnet(L,1)=Runvar(Run,2) THEN 4950
4938     NEXT L
4941     GOTO Abort
4944   END IF
4947   !
4950   Cf_iso=L
4953 Takeup_cf:Where=4
4956   L=Cf_iso
4959   IF NOT Repeat_run THEN
4962     Barrel_wiggled=0
4965     R=100-90*Rb_like
4968     !
4971     ! take up center-filament to target-current
4974     IF Preheated(Sample,1) THEN CALL Filament(2*R,Preheated(Sample,1),.5,1)
4977     CALL Filament(R,Runvar(Run,4),.4,1)
4980     IF Subflag AND Barrel_wiggled THEN Abort
4983     IF Subflag AND NOT Barrel_wiggled THEN
4986       Wiggle_barrel(Barrel_position,Nfils,I_t,Flag(*),Mm$(*))
4989       IF FNFiltest(Nfils,Flag(1))=0 THEN Abort
4992       GOTO 4977
4995     END IF
4998   END IF
5001 Tuneup_cfonly:Where=5
5004   IF Runvar(Run,2) AND Runvar(Run,3) THEN  ! work with focusing isotope
5007     IF Rb_like THEN   ! a triple-filament rubidium-like run; take up cf to get specified isotope beam from turned-off sides
5010       ! wait 1 minute, get a (non-Re) beam by increasing the center-filament only (sides only)
5013       Wait(TIMEDATE,60,Filament(1),Magnet(Cf_iso,1),Auto,Full_auto)
5016       REPEAT ! center-focus on beam, adjust to required beamsize, pausing
5019            ! every 0.3 amps to re-center & re-focus.
5022         Center_peak(0,Cf_iso)    ! center on focusing isotope
5025         Focus(3,2,1,Focnum)
5028         ! (Rb-like triple):  get the minimum center-filament-only Rb beam (or up to 2 times the specified beam)
5031         Size(Runvar(Run,3),Runvar(Run,3)*2,0,Filament(1)+.3,Filament(1)+.3,1,2,30)
5034       UNTIL Subflag<2 OR Filament(1)>6.5 OR Filament(1)>1.5*Runvar(1,4)
5037       ! max. permissible CF current is 1.5x initial CF current
5040       IF Subflag>1 THEN Abort
5043     ELSE  ! not a triple-fil, Rb-like run
5046 !
5049       REPEAT
5052         Center_peak(0,Cf_iso) ! center on the center-fil-only peak (Re-187)
5055         Focus(3-2*(Repeat_run=1),2,1,Focnum)
5058         Size(Runvar(Run,3)/1.1,Runvar(Run,3)*1.1,0,Filament(1)+.3,Filament(1)+.3,1,10,5)  ! adjust the cf-only beam size to wit
hin +/- 10% of spec.
5061       UNTIL Subflag<2 OR Filament(1)>6.5
5064       IF Subflag>1 THEN Abort
5067       Focus(3,2,1,0)
5070     END IF
5073   END IF
5076 END IF
5079    !
5082 S=0 ! calculate sum of isotope-values to see if a Pb-4-6-7-78 run
5085 FOR I=1 TO 8
5088   S=S+Run_iso(Run,I)
5091 NEXT I
5094 L=Rf  ! work with reference-isotope
```

```
5277       Order(N)=I    ! order of data-taking isotopes in Magnet array
5280       GOTO 5289
5283     END IF
5286   NEXT J
5289 NEXT I
5292 RETURN
5295  !
5298 LOOP
5301   CALL Rough(1,Peak(*),E,Data_daly,N,Order(*))
5304   IF E THEN ON 1+Auto GOTO Bmc,Recover_bmc
5307   Rank(Rp,Peak(*),L,N,Mip,Ref,Order(*),Magnet(*))
5310   Center_peak(1,L)
5313   Focus(3,2,1,Focnum)
5316   Enter_beam(Counts,Mv,L,1,0)
5319   IF Mv>10 THEN
5322     CALL Center_barrel(Sample)
5325     IF Subflag=1 THEN Abort
5328   END IF
5331   GRAPHICS OFF
5334 EXIT IF Subflag=0 OR Subflag=4
5337   IF Filament(Nfils)>Runvar(Run,13) THEN Abort
5340   CALL Filament(100-99*Rb_like,Filament(Nfils)+.1-.05*Rb_like,.25,Nfils)  ! if no beam, raise filament current & try again
5343 END LOOP
5346 !
5349 CALL Rough(1,Peak(*),E,Data_daly,N,Order(*))
5352 IF E THEN ON 1+Auto GOTO Bmc,Recover_bmc
5355 Rank(Rp,Peak(*),L,N,Mip,Ref,Order(*),Magnet(*))
5358 Center_peak(1,L)
5361  !
5364 Retake: ! Pre-data-taking beam tuneup, all blocks
5367 Where=12
5370 Goer=1
5373 I_t=1
5376 Abort_count=0
5379 Target=Filament(Nfils)+(Block>0)*Runvar(Run,18)*(Decay<Runvar(Run,23))  ! new sample-fil. current
5382 L=Mip  ! work with the most-intense peak for beam tune-up & beamsize checks
5385 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,1,Coarsemag(1),Foc(*))
5388 Ff=Filament(Nfils)
5391 GOSUB Crtlabel
5394 IF ABS(Target-Ff)>.0001 THEN CALL Filament(2,Target,.25,Nfils)
5397 GOSUB Checkflags
5400 GOTO Check_re
5403 !
5406 Checkflags:FOR I=1 TO 3  ! check flags 3 times in case of GPIO dropouts
5409   CALL Flag(Flag(*),Mu)
5412   Ok=FNFiltest(Nfils,Flag(1))
5415   IF Ok THEN
5418     Fail_checkflags=0
5421     RETURN
5424   END IF
5427   WAIT 2               ! wait 2 seconds before next check
5430 NEXT I
5433 PRINTER IS Prtr(2)
5436 PRINT USING "5/,13X,K,5/";"***** FAILED 'CHECKFLAGS' FILAMENT-FLAG TEST *****"
5439 PRINTER IS CRT
5442 Fail_checkflags=1
5445 GOTO Abort
5448 !
5451 Check_re:IF (Nfils=2) AND (Runvar(Run,2)=187) THEN
5454   Where=13
```

```
5097 Pb_4678=(Ref=206)*(S=825)   ! Pb_4678=1 if a Pb-206/207/208/204 run
5100 Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),0,1,Coarsemag(1),Foc(*))
5103  !
5106 Where=6
5109 Curr1:Barrel_wiggled=0
5112 GRAPHICS OFF
5115 IF Runvar(Run,6)<Filament(Nfils) THEN Curr2
5118 IF Runvar(Run,6)>1.5 AND Runvar(Run,7)<30 THEN CALL Filament(30,.7,1,Nfils) ! take up to .7 amps rapidly if target>1.5 amps
5121 IF Preheated(Sample,Nfils)>Filament(Nfils) THEN CALL Filament(2*Runvar(Run,7),Preheated(Sample,Nfils),.4,Nfils)
5124 Where=7
5127 Curr_1a:CALL Filament((Runvar(Run,7)),Runvar(Run,6),.25,Nfils)   ! first target-current for sample filament(s)
5130 IF Subflag AND Barrel_wiggled THEN Abort
5133 IF Subflag AND NOT Barrel_wiggled THEN
5136   Wiggle_barrel(Barrel_position,Nfils,I_t,Flag(*),Mm$(*))
5139   IF FNFiltest(Nfils,Flag(1))=0 THEN Abort
5142   Barrel_wiggled=1
5145   GOTO 5127
5148 END IF
5151 Start_wait=TIMEDATE
5154  !
5157 Where=8
5160 Wait1:Wait(Start_wait,Runvar(Run,8)*60,Filament(Nfils),Magnet(L,1),Auto,Full_auto)
5163       !
5166 Where=9
5169 Curr2:Barrel_wiggled=0
5172 IF Runvar(Run,9)<Filament(Nfils) THEN Start
5175 CALL Filament((Runvar(Run,10)),Runvar(Run,9),.25,Nfils)   ! second target-current for sample filament(s)
5178 IF Subflag AND Barrel_wiggled THEN Abort
5181 IF Subflag AND NOT Barrel_wiggled THEN
5184   Wiggle_barrel(Barrel_position,Nfils,I_t,Flag(*),Mm$(*))
5187   IF FNFiltest(Nfils,Flag(1))=0 THEN Abort
5190   Barrel_wiggled=1
5193   GOTO 5175
5196 END IF
5199 Start_wait=TIMEDATE
5202  !
5205 Where=10
5208 Wait2:Wait(Start_wait,Runvar(Run,11)*60,Filament(Nfils),Magnet(L,1),Auto,Full_auto)
5211 !
5214 !
5217 Start:Where=11   ! filaments at running-currents, start tuning-up beam
5220 GOSUB Mstart
5223 IF N<2 THEN
5226   FOR I=1 TO 2
5229     PRINTER IS Prtr(I)
5232     PRINT USING "10/,6X,K,8/";"   PROGRAM ERROR - INVALID ISOTOPES PASSED FROM RUN VARIABLES   "
5235   NEXT I
5238   PRINTER IS CRT
5241   Superduperclunk
5244   GOTO Abort
5247 END IF
5250 GOTO 5298
5253  !
5256 Mstart:Good_blocks=0
5259 Verygood_blocks=0
5262 N=0
5265 FOR I=1 TO Niso   ! calculate number & order of data-taking isotopes in the A-array
5268   FOR J=1 TO 8
5271     IF (Run_iso(Run,J)<>0) AND (Magnet(I,1)=Run_iso(Run,J)) THEN
5274       N=N+1
```

```
5628   M=Mu
5631   Mu=M*Daly_ok(Order(K))
5634   OUTPUT 8;Mm$(Mu,I_t)
5637   IF Mu<M THEN WAIT 6
5640   Center_peak((K=1),Order(K))
5643   IF Subflag=4 AND NOT Auto THEN Skey_bmc
5646   IF Subflag=1 AND Auto AND K=1 THEN GOSUB Checkflags
5649 Badcenter:IF Subflag=1 AND NOT Auto THEN Uncentered_pks(K)=1       ! couldn't center on Kth isotope
5652   IF NOT Auto AND K=1 AND (Blocknumber/4=INT(Blocknumber/4) OR Focnum=0) THEN CALL Focus(2,1,1,Focnum)  ! focus every 4th block
 or if unfocused
5655   IF Beam_window AND NOT Auto AND (K=1) AND (Blocknumber>1) THEN
5658     L=Mip  ! restrict beamsize to specified window (manual only)
5661     Size(Wmin,Wmax,0,6,(Fmax),Nfils,10,15)
5664     IF Subflag>1 THEN Abort
5667   END IF
5670   IF K<>1 AND Subflag AND Auto THEN
5673     PRINT Nuclide$(Order(K));Magnet(Order(K),1);"NOT EXPLICITLY CENTERED";CHR$(13)&CHR$(10)
5676     Magnet(K,2)=Magnet0(K)-Magnet0(Mip)+Magnet(Mip,2)
5679   END IF
5682 NEXT K
5685  !
5688 IF NOT Auto AND SUM(Uncentered_pks)=N THEN
5691   PRINT USING "2/,K,2/";FNB1$("*********CAN'T CENTER ANY PEAKS**********")
5694   Superclunk
5697   GOTO Bmc
5700 END IF
5703    !
5706 Scan_all=Auto*(Block(2)+(NOT Auto)*((Roughscan_done=0) OR (SUM(Peak)=0)) !=0 if rough ratios of these isotopes are already know
n
5709 CALL Rough(Scan_all,Peak(*),E,Data_daly,N,Order(*))
5712 IF E THEN ON 1+Auto GOTO Bmc,Recover_bmc
5715 FOR I=1 TO 8
5718   Lastblock_iso(I)=Run_iso(Run,I)
5721 NEXT I
5724 IF Scan_all THEN  ! calculate rough ratios
5727   Rank(Rp,Peak(*),L,N,Mip,Ref,Order(*),Magnet(*))
5730   FOR I=1 TO N
5733     IF I>1 THEN Pkswitch_ratio(I-1)=Peak(1)/Peak(I)
5736     Data_iso(I)=Magnet(Order(I),1)
5739   NEXT I
5742   IF Pb_4678 AND Data_iso(4)<>204 THEN Pb_4678=0 ! don't treat as natural-Pb run if 204 isn't the least-intense peak
5745 ELSE  ! calculate beamsizes for all peaks using first rough-scan ratios
5748   FOR I=2 TO N
5751     Peak(I)=Peak(1)/Pkswitch_ratio(I-1)
5754   NEXT I
5757 END IF
5760 IF Peak(1)<((Daly=0)*10+(Daly>0)*2) THEN ! if MIP-center didn't recalibrate mag-settings, normalize mag-settings for all other
isotopes to MIP
5763   FOR I=1 TO Niso
5766     IF I<>Mip THEN Magnet(I,2)=Magnet0(I)+Magnet(Mip,2)-Magnet0(Mip)
5769   NEXT I
5772 END IF
5775  !
5778 Take_data:Where=17  ! take isotope-ratio data
5781 GOSUB Printtime
5784 Valid_block=Block
5787 IF Fil_in=Filament(Nfils) AND Data_daly=Daly_out AND NOT Pb_4678 AND (TIMEDATE-Blockend_t(240) AND ((NOT Auto)*(Blocknumber>1)
OR Auto*(Block>0)) THEN
5790   Share_bkgrds=1
5793 ELSE
```

```
5457 Re1:L=0    ! adjust size of Re-187 beam using center-filament
5460   Correct(Ff$(*),Mm$(*),Filament(*),Magnet(0,2),Mu,1,Coarsemag(0),Foc(*))
5463   WAIT 2
5466   Size(Runvar(Run,3)/1.1,Runvar(Run,3)*1.1,0,10,6,1,10,15)
5469   IF Subflag>1 THEN Abort
5472   Correct(Ff$(*),Mm$(*),Filament(*),Magnet(Mip,2),Mu,1,Coarsemag(1),Foc(*))
5475 END IF
5478 WAIT 2
5481 !
5484 Sf1:L=Mip   ! adjust sample-fil-currents for specified beam
5487 Where=14
5490 IF Block=0 THEN
5493   REPEAT
5496     ! beam-size adjust before 1st-block:  re-check for MIP every .1 amp, in
5499     ! case of large isobaric interferences
5502     Size(Runvar(Run,14),Runvar(Run,15),ABS(Runvar(Run,17)),Runvar(Run,16),Filament(Nfils)+.1,Nfils,(Runvar(Run,10)),10)
5505     Sizeflag=Subflag
5506     IF Sizeflag>2 THEN Abort
5508     CALL Rough(1,Peak(*),E,Data_daly,N,Order(*))
5511     IF E THEN Recover_bmc
5514     Rank(Rp,Peak(*),L,N,Mip,Ref,Order(*),Magnet(*))
5517   UNTIL Sizeflag(2 OR Filament(Nfils))=Runvar(Run,13)
5520   IF Sizeflag>1 THEN Abort
5523 ELSE
5526   IF Reduced_current=0 THEN CALL Size(Runvar(Run,14),Runvar(Run,15),ABS(Runvar(Run,17)),Runvar(Run,16),Runvar(Run,13),Nfils,(Ru
nvar(Run,10)),10)
5529 ! Beam-size adjust after 1st-block
5532   IF Subflag>1 THEN Abort
5535 END IF
5538 Subflag=0
5541 Start_wait=TIMEDATE
5544 IF Block THEN CALL Center_peak(1,L)
5547 CALL Decay(Reduced_current,Flash,Runvar(Run,14),Runvar(Run,23),12,Block)
5550 IF Subflag THEN Abort
5553 IF Block(2 OR Flash OR Block/4=INT(Block/4) THEN  ! focus 1st block, every 4th block & after flash
5556   Center_peak(1,L)
5559   Focus(3,1,1,Focnum)
5562 END IF
5565 IF Block/6=INT(Block/6) THEN
5568   Center_barrel(Sample) ! center barrel every 6th block
5571   IF Subflag=1 THEN Abort
5574 END IF
5577 IF ABS(Ff-Filament(Nfils))(.0001 OR Block=0 THEN
5580   IF Reduced_current=0 THEN CALL Size(Runvar(Run,14),Runvar(Run,15),ABS(Runvar(Run,17)),Runvar(Run,16),Runvar(Run,13),Nfils,(Ru
nvar(Run,10)),10)
5583   IF Subflag>1 THEN Abort
5586 END IF
5589 !
5592 IF Block=0 THEN
5595 Qwait:Where=15                !"DATA-WAIT"
5598   IF Runvar(Run,12) THEN CALL Wait(Start_wait,Runvar(Run,12)*60,Filament(Nfils),Magnet(L,1),Auto,Full_auto)
5601 END IF
5604 !
5607 Go:Where=16   ! final peak-centering & quick scan before data-taking block
5610 GRAPHICS OFF
5613 Nsets=Runvar(Run,22)
5616 !
5619 Mgo:MAT Uncentered_pks= (0)
5622 Mip=Order(1)
5625 FOR K=2-(NOT Block) OR (NOT Auto) TO N
```

```
5964   IF Aver(I)>0 THEN        ! protect against zero-to-power error
5967     IF Aver(I)^Inv<Minrat THEN Minrat=Aver(I)^Inv
5970   END IF
5973 NEXT I
5976 IF Minrat<1 THEN Mipbeam=Pk/Minrat
5979 IF Minrat>=1 THEN Mipbeam=Pk
5982 IF Mipbeam>Runvar(Run,14) THEN Verygood_blocks=1+Verygood_blocks ! #blocks with good sigmas at MIP>MINBEAM
5985 !
5988 OUTPUT 8 USING "4A,4Z";"$OMV",Good_blocks  ! display # of good blocks on the sysmon
5991 FOR I=1 TO Good_blocks ! tootle once for each good block
5994   IF I>1 THEN WAIT .07
5997   BEEP 660,.05
6000   BEEP 720,.05
6003   BEEP 660,.05
6006 NEXT I
6009 IF Block<Runvar(Run,20) THEN  ! haven't reached MAX# BLOCKS yet
6012   IF (Good_blocks<Runvar(Run,19)) THEN Retake ! take more blocks if required
6015   IF Verygood_blocks<Runvar(Run,19) AND Runvar(Run,17)<0 THEN  ! if default beam <0, then try and get good blocks with beam>min
beam
6018     Runvar(Run,17)=Runvar(Run,14)  !  default-beam = min beam
6021     Good_blocks=Verygood_blocks    !  reset good-block counter
6024     GOTO Retake
6027   END IF
6030 END IF
6033 !
6036 Quit:   ! finished with data-taking for this run
6039 IF Block=0 AND NOT Outgas THEN Abort
6042 IF Block>1 AND NOT Outgas THEN
6045   Printres(Prtr(2),Run,E)    ! print out results for this run
6048   Average(1,Normal(*),Candump,E,(Run),Prtr(*)) ! calculate wtd averages for this run
6051 END IF
6054  !
6057 Do_next_run:IF Run=32 THEN Finish      ! 32 runs max.
6060 IF Run_order(1+Run)=0 THEN Finish      ! if no next-run defined
6063 Repeat_run=(Run_order(Run+1)=Sample)   !=1 if next run will re-use this sample
6066 IF Repeat_run THEN Runvar(1+Run,9)=MAX((Filament(Nfils)),(Runvar(1+Run,9)))    ! don't decrease fil-currents if reusing same sa
mple
6069 Run=1+Run  ! go on to next run
6072 IF NOT Repeat_run THEN CALL Zero_fils(Filament(*))
6075 GOTO Autcont
6078  !
6081  !
6084 Abort:  ! If flags OK, abort this run, print out machine-status data, & go               to next run.  But first, put 0.2 amps
 through any fils with
6087         ! less than 0.2 amps, & check the filament-flags.
6090 GRAPHICS OFF
6093 MAT Orig_fil= Filament
6096 FOR I=0 TO 1
6099   FOR J=1 TO 2
6102     IF Filament(I+2*J-1)<.2 THEN OUTPUT 8;Ff$(I,J),200
6105   NEXT J
6108 NEXT I
6109 IF NOT Fail_checkflags THEN
6110   CALL Flag(Flag(*),Mu)
6111   WAIT 2
6112 END IF
6114 Enter_beam(0,Mu,1,I_t,1)
6117 IF Mu>5 AND Fail_checkflags THEN ON 1+(Where=12)+2*(Where=16) GOTO 6120,Check_re,Badcenter
6120 IF FNFiltest(Nfils,Flag(1))=0 THEN
6123   Badflag=1+Badflag
```

```
5796   Share_bkgrds=0
5799 END IF
5802  ! share backgrounds between blocks if 1) filament-currents remain unchanged, 2) collector is the same, 3) no more than 4 minut
es have elapsed
5805  !
5808 Start_block=TIMEDATE ! starting-time of block
5811 Fil_in=Filament(Nfils)
5814 !
5817 ! Take isotope-ratio data
5820 Data(Decay,Run_name$(Run),Ln,Blocknumber,Numberofblocks,Share_bkgrds,Dump_datagraf,Bad_pressure,Niso,Block,Data_daly,Pb_4678,Ru
n)
5823 !
5826 IF Bmc_out THEN Recover_bmc
5829 IF Next_run THEN Recover_quit
5832 IF NOT Auto AND Bad_pressure THEN Bmc
5835 Blockend_t=TIMEDATE
5838 Block_time=((Blockend_t+Start_block)/2-Start_time)/60 ! "time" of block, in minutes
5841 GOSUB Printtime
5844 Daly_out=Data_daly
5847 Reduced_current=0
5850 IF NOT Spike THEN PRINT USING "2/"
5853 PRINTER IS CRT
5856 IF Block=Valid_block THEN ON Auto+1 GOTO Bmc,Resume ! If didn't complete block
5859 FOR I=N TO 7
5862   Acc(I)=0
5865   Sigma$(I)=""
5868   Delta$(I)=""
5871   Ratio$(I)=""
5874   Aver(I)=0
5877 NEXT I
5880 Nn=N+(Spike<>0)*(1+(Spikedrun_iso(3)<>0)) ! add ratios if a spiked run
5883 IF Spike THEN CALL Spikecorr(Aver(*),Last_aver(*),Ratio$(*),Acc(*),Lacc(*),Last_ratio$(*),Delta$(*),Prtr(*),(N),Block)
5886 Writedata(Aver(*),Ratio$(*),Ffile,Run_name$(Run),Sigma$(*),Delta$(*),Acc(*),Pk,Block_time,Filament(Nfils),Block,Run,Sample,(Nn)
,Data_collector,Prtr(*))
5889 IF NOT Auto AND Limit_growth AND (Decay>Maxgrowth) THEN CALL Reduce(0,(Maxgrowth),Filament(*),Runvar(Run,19),Nfils)
5892 ! max. permissible beam-growth is 1x 'MAX GROWTH' for BEAM>MINBEAM &Block)1;        2x for DEFBEAM<BEAM<MINBEAM &Block)1; 3x f
or BEAM<DEFBEAM
5895 IF Auto AND ((Pk)=ABS(Runvar(Run,17))) OR (Decay)=3*Runvar(Run,23))) THEN
5898   IF (Decay)2*Runvar(Run,23)) OR (Decay)Runvar(Run,23))*(Pk)Runvar(Run,14)) THEN
5901     Reduce(Reduced_current,Runvar(Run,23),Filament(*),Runvar(Run,18),Nfils)
5904   END IF
5907 END IF
5910  !
5913  !
5916 Resume:IF (Block=0) AND Auto THEN Retake
5919 IF NOT Auto THEN RETURN
5922 !
5925 FOR J=1 TO N-1  ! test to see if precision or ratios for this block is ok
5928   Sigma=FNSigpar(Sigma$(J)) ! returns sigma%, negative if w/in theor.limits
5931   S=Runvar(Run,21)          ! max. acceptable sigmamean%
5934   IF S)=0 THEN              ! theoretical limits OK in all cases
5937     IF S(Acc(J) AND Sigma)0 THEN 5988
5940   ELSE                     ! theoretical limits only OK if within tolerance
5943     IF ABS(S)(Acc(J) THEN 5988
5946   END IF
5949 NEXT J
5952 !
5955 Good_blocks=Good_blocks+1  != # blocks with OK sigmas so far
5958 Minrat=9.E+99              ! find size of MIP
5961 FOR I=1 TO N
```

```
"
6300 PRINT USING "20X,3(K),17/";FNClock_12$(TIME$(TIMEDATE)),",   ",DATE$(TIMEDATE)
6303 PRINTER IS CRT
6306 GRAPHICS OFF
6309 PRINT USING "18/,241A,/,K,/,240A,4/";""&RPT$("*",240),RPT$("*",25)&"  AUTOMATIC RUNNING FINISHED  "&RPT$("*",25),RPT$("*",240)
&""
6312 FOR I=1 TO 10
6315   Whoop
6318 NEXT I
6321 Mu=0
6324 I_t=2
6327 OUTPUT 8;Mm$(0,2)  ! switch to cup, integration-time of 0.2 seconds
6330 CALL Filament(1000,0,0,1)  ! turn off center-fil
6333 CALL Filament(1000,0,0,2)  !  "   "   side   "
6336 DISP                       ! to delete invalid fil-curr display
6339 IF Run>1 THEN GOSUB Result
6342 Auto=0
6345 Full_auto=0
6348 PRINT USING "18/,241A,/,K,/,241A,8/";""&RPT$("*",240),RPT$("*",25)&"  AUTOMATIC RUNNING FINISHED  "&RPT$("*",25),RPT$("*",240)
&""
6351 CALL Pattern
6354 GOTO Bmc
6357  !
6360 Crtlabel:PRINTER IS CRT
6363 OUTPUT 2;Clear$;
6366 PRINT USING "9A,2D,45A,15X,6A,2D,K,/";"BARREL#"&Ci$&" ",Sample," "&Cn$&"     "&Run_name$(Run)[1,40],"BLOCK#"&Ci$,1+Block,Cn$
6369 RETURN
6372 !
6375 Printtime:PRINTER IS Prtr(2);WIDTH (80)
6378 PRINT USING "K,K";CHR$(27)&"&k0S",RPT$("X",27)&" ";FNClock_12$(TIME$(TIMEDATE))&"  "&DATE$(TIMEDATE)&" "&RPT$("X",28)
6381 RETURN
6384 !
6387 Off_preheat: ! turn preheat filaments down to zero
6390 IF Filament(3) OR Filament(4) THEN PRINT USING "5/,6X,K,2/";CHR$(130)&"PREHEAT-FILAMENTS HAVE BEEN ON FOR TWO HOURS--  TURNING
 OFF"&CHR$(128)
6393 IF Filament(3) THEN CALL Filament(20,0,0,3)  ! center-preheat
6396 IF Filament(4) THEN CALL Filament(20,0,0,4)  ! side-preheat
6399 RETURN
6402 !
6405 Retrn:RETURN  ! Dummy RETURN
6408 Clearbmc:GOSUB Clearall
6411 GOTO Bmc_1
6414 END
6417 !
6420 !
6423 Endmain: ! ************ END OF MAIN PROGRAM ******************************
6426 Center:SUB Center_peak(Recalibrate,INTEGER C)
6429   OPTION BASE 1
6432    !
6435    !  Find magnet position for center of peak
6438    !
6441   COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t0
6444   COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
6447   COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
6450   COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
6453   COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
6456   COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
6459   INTEGER B,Base,Pr
6462   REAL Peakside(2)
6465   DATA 0,0,0,0
```

```
6468   READ Number_tries,Subflag,Daly_tested,Escape
6471   FOR I=2*Auto*Full_auto TO 19
6474     ON KEY I LABEL "" CALL Clunk
6477   NEXT I
6480   OFF KBD
6483   OFF KNOB
6486   ON KEY 9 LABEL "    ESCAPE" GOTO Escape_center
6489   L=C  !  order of peak in the Magnet-array
6492   PRINT "CENTERING ";CHR$(129);Magnet(L,1);CHR$(128),"(starting value is ";VAL$(Magnet(L,2))&")"
6495   M=Mu
6498   As=Aside*(L<>0)+37*(NOT L)  ! if Re-187, use a half-peak offset of 37
6501   Mu=Mu*Daly_ok(L)
6504   Lx1=Magnet(L,2)
6507   Lx2=Lx1
6510   B=Magnet(L,2)
6513   IMAGE "$DFJ",4Z
6516   !
6519   Correct(Ff$(*),Mm$(*),Filament(*),B,Mu,2,Coarsemag(L<>0),Foc(*))
6522   IF Mu<M THEN WAIT 6
6525   FOR I=1 TO 2
6528     Enter_beam(X,Mv,L,2,Pr)
6531     IF Pr>1 THEN 6519
6534   NEXT I
6537   IF NOT Daly_tested AND NOT Mu AND Mv<10 AND Daly THEN
6540     Mtest
6543     Daly_tested=1
6546     GOTO 6525
6549   END IF
6552   Number_readings=1+5*((Mu=0)*((Mv(10)+(Mv(3)*(Mv(1))+Mu*((Mv(3)+(Mv(1)+(Mv(.2))) ! number of .2-sec integrations on each side
of peak
6555   OUTPUT 8;Mm$(Mu,2)
6558   Centered=0
6561   !
6564   REPEAT
6567     MAT Peakside= (0)
6570     FOR J=1 TO 2
6573       OUTPUT 8 USING 6513;B+(2*J-3)*As
6576       FOR I=1 TO Number_readings+1
6579         Enter_beam(X,Mv,L,2,Pr)
6582         IF Pr>1 THEN 6519
6585         IF I>1 THEN Peakside(J)=Peakside(J)+Mv/Number_readings
6588       NEXT I
6591       PRINT DROUND(Peakside(J),3);
6594     NEXT J
6597     Number_tries=Number_tries+1
6600     Toler=(Mu=0)*.2+Mu*.006
6603     IF (Number_tries>30) OR ((Peakside(1)<Toler) AND (Peakside(2)<Toler)) THEN
6606       PRINT USING "2/,3(K)";"**** CAN'T CENTER ",Magnet(L,1)," ****"
6609       IF Full_auto THEN CALL Clunk
6612       Subflag=1
6615       SUBEXIT
6618     END IF
6621   !
6624     Sumsides=Peakside(1)+Peakside(2)
6627     Diffsides=Peakside(2)-Peakside(1)+1.E-9
6630     IF ABS(Diffsides)<.018*(1+Number_readings)*Sumsides THEN Centered=1
6633   !
6636     Stepmult=1+1.5*(Sumsides)<3*(Mu=0)+.4*Mu) AND (Sumsides/ABS(Diffsides)-1<.02))
6639     ! increase hunting step if  difference in peakside intensities is large
6642     Z=Stepmult*.21*As*(Diffsides)/(Sumsides+(NOT Sumsides)) !  .21 is "hunting" rate
```

```
6645     B=B+Z*(ABS(Z))=1)+SGN(Z)*(ABS(Z)<1)
6648     IF B=Lx2 THEN Centered=1
6651 !
6654     Lx2=Lx1
6657     Lx1=B
6660     PRINT TAB(37);B
6663   UNTIL Centered
6666   !
6669   PRINT TAB(37);Ci$;B;Cn$;CHR$(13)&CHR$(10)
6672   Magnet(L,2)=B
6675   IF Recalibrate AND L AND ((Mu=0)*(Mv>10) OR Mu*(Mv>2)) THEN
6678     ! if good beam, use new mag-value to estimate mag-setting of all peaks
6681     Mcoef(1)=Mcoef(1)+(B-FNIsomag(Mcoef(*),(Magnet(L,1))))
6684     FOR I=1 TO 24
6687       M=Magnet(I,1)
6690       IF M THEN Magnet(I,2)=Mcoef(1)+M*(Mcoef(2)+Mcoef(3)*M)
6693     NEXT I
6696   END IF
6699 Done:Correct(Ff$(*),Mm$(*),Filament(*),B,Mu,I_t0,Coarsemag(L<>0),Foc(*))
6702   IF NOT Auto THEN BEEP 500,.03
6705   SUBEXIT
6708   !
6711 Escape_center:Subflag=4
6714   GOTO Done
6717 SUBEND  ! ------------------------------------------------------------
6720 !
6723!
6726 Focus:SUB Focus(Max_cycles,Min_jump,Quick,Foc_number)   ! do automatic ion-optics focusing
6729   OPTION BASE 1
6732   COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t0
6735   COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
6738   COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
6741   COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
6744   COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
6747  !
6750   INTEGER Tempfoc(8),Lastjump,Lastjump_but1,Lastjump_but2,I,Pr,C,Use(8),I_t
6753   DIM Peak(3)
6756   DATA 1,1,1,1,1,1,1,1,64,20,0
6759   GCLEAR
6762   OFF KNOB
6765   READ Use(*),Offset0,Maxtry,Subflag
6768   IF Foc_number=0 THEN First_focus=1
6771   FOR I=Auto*Full_auto*2 TO 19
6774     ON KEY I LABEL "" CALL Clunk
6777   NEXT I
6780   OFF KBD
6783   OUTPUT KBD;CHR$(255)&CHR$(75);
6786   ON KEY 9 LABEL "    ESCAPE" GOTO 7428
6789   M=Daly_ok(L)
6792   MAT Daly_ok= (0)
6795   Daly_ok(L)=M
6798   C=Coarsemag(L<>0)
6801   PRINTER IS CRT
6804   IMAGE "$0F",4A,4Z
6807   MAT Tempfoc= Foc
6810   CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,2,C,Foc(*))
6813   CALL Enter_beam(Count,Mv,L,2,1)
6816   I_t=(Mu<10)+(Mu<40)
6819   IF Mu<20 AND NOT Mu AND Daly THEN
6822     CALL Mtest
```

```
6825    IF Mu=0 THEN WAIT 6
6828    GOTO 6813
6831  END IF
6834  IF Mu*(Mv<.02) OR (NOT M)*(Mv<.8) THEN    ! Scan trigger at .02 mV/.8mV
6837   !
6840    Mu=(Daly)0)
6843    CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,2,C,Foc(*))
6846    Scan=1     ! if no apparent beam, scan each plate until a beam is found
6849    DATA 3,6,4,5,2,7,1,8   ! order of plates for panic-scan
6852    ON KEY 4 LABEL "  SKIP PLATE" GOTO 6894
6855    RESTORE 6849
6858    FOR I=1 TO 6+Nfils
6861      READ Plate
6864      IF Use(Plate) THEN
6867        FOR J=460*(Plate=8) TO 999-459*(Plate=8) STEP 10-6*(Plate=1)-8*(Plate=8)
6870          OUTPUT 8 USING 6804;Z$[10-Plate,10-Plate],J
6873          CALL Enter_beam(X,Mv,L,2,Pr)
6876          DISP "SCANNING TO FIND BEAM-   PLATE";Plate;"=";J;TAB(45);"BEAM =";DROUND(Mv,2);"mV"
6879          ON Pr GOTO 6882,6942,6810
6882          IF Mu*(Mv)).02) OR (NOT Mu)*(Mv)).8) THEN 6912
6885        NEXT J
6888        OUTPUT 8 USING 6804;Z$[10-Plate,10-Plate],Tempfoc(Plate)
6891      END IF
6894    NEXT I
6897    PRINT USING "/,K,2/";FNH$("UNABLE TO FIND BEAM")
6900    Subflag=1
6903    Clunk
6906    SUBEXIT
6909       !
6912    Foc(Plate)=J+10-6*(Plate=1)  !  found a beam during the panic-scan
6915    Tempfoc(Plate)=Foc(Plate)
6918  END IF
6921   !
6924  IF First_focus THEN
6927    Graph_ymax=4
6930    Ymax=10000
6933    Graph_ymin=-2*Mu
6936    Xmax=600
6939  ELSE
6942    Graph_ymax=Mv*(1.3+.5*(Foc_number(2))
6945    Ymax=Graph_ymax
6948    IF Graph_ymax)1.E+4 THEN Graph_ymax=1.E+4
6951    Xmax=300
6954    Graph_ymin=0
6957  END IF
6960  Y=Mv
6963  CALL Axes(20,100,8,100,0,Xmax,Graph_ymin,Graph_ymax,"STEP (ION OPTICS FOCUS)","mV "&VAL$(Magnet(L,1)),0,Mu,(First_focus))
6966  Foc_number=1+Foc_number
6969  CSIZE 2.8
6972 !
6975  FOR Cycle=1 TO Max_cycles
6978    Change=0
6981    Foctime=Foctime+10
6984    Mm=Mu
6987    IF Cycle)1 OR Scan THEN
6990      CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,C,Foc(*))
6993      CALL Enter_beam(X,Mv,L,I_t,1)
6996      IF NOT First_focus AND Mv)1.2*Ymax THEN
6999        Ymax=2*Mv
7002        Graph_ymax=Ymax
```

```
7005        IF Ymax>1.E+4 THEN Ymax=1.E+4
7008        Foctime=0
7011        Xmax=Xmax*.7
7014        CALL Axes(20,100,8,100,0,Xmax,0,Ymax,"CYCLE","mV BEAM",0,Mu,0)  ! draw plot-box for graphics display
7017       END IF
7020     END IF
7023     Offset0=Offset0/(1+(Offset0>16))
7026     Min_jump=Min_jump/(1+(Min_jump>1)) !  resolution, in focus units
7029     I_t=1+Mu*(Mv>3)+(NOT Mu)*(Mv>15)
7032     IF Mu AND Daly AND (P>40) THEN
7035       Mu=0
7038       Daly_ok(L)=0
7041     END IF
7044     CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,C,Foc(*))
7047     IF Mm<>Mu THEN WAIT 6
7050     ON KEY 4 LABEL "  SKIP PLATE" GOTO Plate_done
7053      !
7056 Plates: ! optimize beam for each focusing-potential
7059     FOR Plate=1 TO 6+Nfils
7062       IF (Use(Plate)=0) OR (Plate=1)*(Cycle=1) OR (NOT Quick)*((Plate<>1)*(Plate<>2)*(Plate<>4)*(Plate<>8)) THEN Nextplate
7065       Min_value=460*(Plate=8)  ! minimum allowable value for this plate
7068       Max_value=999-459*(Plate=8) ! maximum     "       "     "   "   "
7071       Point_plotted=0
7074       Foctime=Foctime+6          ! graphics X-axis (# changes)
7077       No=Noise(Mu)^2
7080       O$=Z$[10-Plate,10-Plate]
7083       Offset=Offset0
7086       IF (Plate=8) AND (Offset>8) THEN Offset=8
7089       IF Plate>1 THEN CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,C,Foc(*))
7092       DATA 1,1,0,0,0
7095       RESTORE 7092
7098       READ Num_jumps,Jump_dir,Lastjump,Lastjump_but1,Lastjump_but2
7101       MOVE Foctime+2,Y+Graph_ymax/20
7104       PRINT
7107       LABEL Plate
7110       !
7113 Reset_max:DISP FNH$(VAL$(Plate))&"   mV ";TAB(30);"CYCLE";Cycle;"OF";Max_cycles,"CHANGE:";Change
7116       Updown=0
7119       GOSUB Foc_change
7122       GOTO Next_jump
7125      !
7128 Foc_change:OUTPUT 8 USING 6804;O$,Tempfoc(Plate)+Updown*Offset
7131       PRINT VAL$(Tempfoc(Plate)+Updown*Offset);
7134       FOR J=1 TO I_t
7137         CALL Enter_beam(X,Mv,L,I_t,Pr)
7140       NEXT J
7143       ON Pr GOTO 7146,7077,6990
7146       Peak(2+Updown)=Mv
7149       PRINT TAB(6);VAL$(DROUND(Mv,3+(Mv>1000)))
7152       Foctime=Foctime+1
7155       IF First_focus THEN
7158         IF Mv>0 THEN
7161           Y=LGT(Mv)
7164         ELSE
7167           Y=-Mu
7170         END IF
7173       ELSE
7176         Y=Mv
7179       END IF
7182       IF NOT Point_plotted THEN
```

```
6126   IF Badflag=1 THEN  ! re-find sample & try again
6129     FOR Y=2 TO 1 STEP -1
6132       PRINTER IS Prtr(Y)
6135       PRINT USING "3/,3(K),3/";"** NO FILAMENT-CONTACTS (sample ",Sample,") -- TRYING TO RECOVER BY RE-FINDING SAMPLE **"
6138     NEXT Y
6141     Superclunk
6144     Find(0,Sample,Nfils,Subflag,Filament(*),0)
6147     IF Subflag THEN Give_up
6150     !
6153     FOR I=1 TO 4   ! take up fil-currents to initial values
6156       IF (I=1 AND Nfils=1) OR (I=2 AND Nfils=2) THEN
6159     ! take up sample-filament rapidly to 85% of orig. curr, then slowly to orig. curr
6162         CALL Filament(200,.85*Orig_fil(I),.5,(I))
6165         CALL Filament(10,Orig_fil(I),.5,(I))
6168       ELSE
6171         IF Fl(I) THEN CALL Filament(200,Orig_fil(I),.5,(I))
6174       END IF
6177     NEXT I
6180     IF (Filament(3)=0) AND (Filament(4)=0) THEN CALL Wait(TIMEDATE,15,Filament(Nfils),Magnet(L,1),Auto,Full_auto)
6183     GOTO Resume_auto
6186   END IF
6189 END IF
6192 !
6195 Give_up:FOR Y=1 TO 2
6198   PRINTER IS Prtr(Y)
6201   PRINT USING "3(K,/)";RPT$("*",80),RPT$("*",80),RPT$("*",80)
6204   PRINT "**** ABORTED RUN FOR BARREL#":Sample:"    RUN#":Run:"****"
6207   PRINT "CENTER FIL.=";Filament(1),"SIDE FIL.=":Filament(2),"TIME IS ";FNClock_12$(TIME$(TIMEDATE))
6210   PRINT DROUND(Mv,2);"mV";Magnet(L,1):"BEAM.      Where=":Where
6213   PRINT USING "K,2/,8(3D,3X),/,8(4D,2X),/":"FOCUSING VALUES:",1,2,3,4,5,6,7,8,Foc(*)
6216   PRINT USING "/,2(K,/),K":RPT$("*",80),RPT$("*",80),RPT$("*",80)
6219   IF Y=1 THEN CALL Superclunk
6222 NEXT Y
6225 GOSUB Key_escape
6228 FOR I=1 TO 40
6231   WAIT .2
6234 NEXT I
6237 IF Block THEN
6240   Printres(Prtr(2),Run,E)
6243   IF Block>1 THEN CALL Average(1,Normal(*),Candump,E,(Run),Prtr(*))
6246 ELSE
6249   Writedata(Aver(*),Ratio$(*),Ffile,Run_name$(Run),Sigma$(*),Delta$(*),Acc(*),Pk,Block_time,Filament(Nfils),Block,Run,Sample,N,
Data_collector,Prtr(*))
6252 END IF
6255 IF NOT Outgas AND Goer=0 AND Mv<2 AND (Filament(Nfils)>Runvar(Run,13) OR Filament(1)>6) THEN Abort_count=Abort_count+1  ! # of
successive no-beam aborts
6258 IF Abort_count<2 OR Just_outgassing THEN Do_next_run
6261 !
6264 FOR Y=2 TO 1 STEP -1
6267   PRINTER IS Prtr(Y)
6270   PRINT USING "10/,K,2/";FNBl$("  ****AUTO OPERATION SUSPENDED -- 2 SUCCESSIVE NO-BEAM ABORTS****  ")
6273   PRINT USING "K,2/";"(press  RECALL  key to resume auto-operation)"
6276 NEXT Y
6279 Superduperclunk
6282 Zero_fils(Filament(*))
6285 GOTO Bmc
6288  !
6291  !
6294 Finish:PRINTER IS Prtr(2)
6297 PRINT USING "K,4/,40A,/,6X,K,/,40A,2/,K";CHR$(27)&"&k1S",RPT$("*",40)," AUTOMATIC RUNNING FINISHED",RPT$("*",40),CHR$(27)&"&k0S
```

```
7185        MOVE Foctime,Y
7188        Point_plotted=1
7191      ELSE
7194        DRAW Foctime,Y
7197      END IF
7200      Num_jumps=Num_jumps+1+(Offset=Min_jump)
7203      IF Num_jumps>Maxtry THEN Plate_done
7206      RETURN
7209       !
7212 Next_jump:A=(Plate<>8)*(FNLimit(Tempfoc(Plate)+2*Jump_dir*Offset,Min_value,Max_value)=0)
7215      B=(Lastjump=Updown*Offset)*(Lastjump_but1=Updown*Offset)*(Lastjump_but2=Updown*Offset)
7218      IF A AND B THEN Offset=2*Offset ! double offset if last 3 offsets stayed the same, in order to converge on true max more
rapidly
7221       !
7224      IF FNLimit(Tempfoc(Plate)+Jump_dir*Offset,Min_value,Max_value) THEN  ! new value not within limits- reduce offset
7227        IF NOT Tracking_incr THEN Change_offdir
7230        Offset=Offset/(1+(Offset>Min_jump))
7233        ON 1+(Offset<=Min_jump) GOTO 7224,Plate_done ! plate focused if can't reduce offset any more
7236      END IF
7239      Updown=Jump_dir
7242      GOSUB Foc_change
7245      Lastjump_but2=Lastjump_but1
7248      Lastjump_but1=Lastjump
7251      Lastjump=Updown*Offset
7254      IF (Peak(2+Updown)>Peak(2)+FNDiff(Peak(2),Peak(2+Updown),No,I_t,Ions)) AND NOT FNLimit(Tempfoc(Plate)+Updown*Offset,Min_v
alue,Max_value) THEN
7257        ! if new plate-setting gives a beam greater that the previous-max + theor. noise, then define as the new max-setting &
beam
7260        Peak(2)=Peak(2+Updown)
7263        Peak(1)=0
7266        Peak(3)=0
7269        Tempfoc(Plate)=Tempfoc(Plate)+Updown*Offset
7272        GOTO Next_jump
7275      END IF
7278      Offset=Offset/(1+(Offset>Min_jump))
7281      IF FNLimit(Tempfoc(Plate)-Jump_dir*Offset,Min_value,Max_value) THEN
7284        ! if proposed new setting isn't within permissible limits of this plate, cut offset in half again
7287        Offset=Offset/(1+(Offset>Min_jump))
7290        ON 1+(Offset<=Min_jump) GOTO Reset_max,Plate_done
7293      END IF
7296      !
7299      Updown=-Jump_dir  ! change offset direction
7302      GOSUB Foc_change
7305      IF (ABS(Peak(3)-Peak(2))<FNDiff(Peak(2),Peak(3),No,I_t,Ions)) AND ABS(Peak(1)-Peak(2)<FNDiff(Peak(2),Peak(1),No,I_t,Ions)
) THEN Plate_done
7308      IF Peak(2+Updown))>Peak(2) THEN
7311 Change_offdir:Tracking_incr=1
7314        Jump_dir=-Jump_dir
7317        IF FNLimit(Tempfoc(Plate)+Updown*Offset,Min_value,Max_value) THEN Plate_done
7320        Tempfoc(Plate)=Tempfoc(Plate)+Updown*Offset
7323        Peak(1)=0
7326        Peak(3)=0
7329        GOTO Next_jump
7332      END IF
7335      IF Peak(2)=0 THEN Reset_max
7338      IF Offset>Min_jump THEN
7341        Offset=Offset/2
7344        GOTO Reset_max
7347      END IF
7350       !
```

```
7353 Plate_done:IF Plate>8 THEN Nextplate
7356       OUTPUT 8 USING 6804;0$,Tempfoc(Plate)
7359       Change=Change+ABS(Foc(Plate)-Tempfoc(Plate)) ! sum of changes for this cycle
7362       Foc(Plate)=Tempfoc(Plate)
7365 Nextplate:NEXT Plate
7368     IF Change<24 THEN 7380
7371     BEEP 1000,.05
7374   NEXT Cycle
7377    !
7380   FOR I=1 TO 7
7383     IF (Foc(I)=0) THEN Lowfoc=Lowfoc+1
7386     IF (Foc(I)=9999) THEN Highfoc=Highfoc+1
7389   NEXT I
7392   IF Lowfoc>1 OR Highfoc>1 OR Lowfoc+Highfoc>2 THEN
7395     PRINT USING "2/,k";FNBl$("***** SUSPECT FOCUS-SETTINGS: POSSIBLE HARDWARE PROBLEMS *****")
7398     GRAPHICS OFF
7401     CALL Beep(160,.03,.03,60)
7404   END IF
7407   CALL Whoop
7410   OUTPUT 8;Mm$(Mu,I_t0)
7413   PRINT TABXY(26,17);
7416   PRINT USING "8(D,4X),K";1,2,3,4,5,6,7,8,"-PLATE"
7419   PRINT USING "24X,8(3D,2X),K";Foc(*)," -SETTING"
7422   DISP
7425   SUBEXIT
7428   Subflag=4*(Auto=0)
7431   Broop
7434   CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t0,C,Foc(*))
7437   GOTO 7410
7440 SUBEND  ! ------------------------------------------------------------
7443 !
7446 !
7449 Barrel:SUB Center_barrel(INTEGER Sample)
7452   OPTION BASE 1
7455   COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t0
7458   COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
7461   COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
7464   COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
7467   COM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel
7470   COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
7473   REAL F1(4),Barrel_pk(100),P1(2),P2(2),Flg(2)
7476   INTEGER I,Pr,B,I_t,Bmin,F_subflag
7479   DATA 0,0,0,0
7482   READ Subflag,Find_flag,F_acheck,F_bcheck ! zero flags
7485   !
7488   IF Filament(1)<.2 AND Filament(2)<.2 THEN
7491     PRINT USING "10/,14X,K,6/";FNH$("SORRY, CAN'T CENTER THE BARREL WITHOUT A BEAM")
7494     Clunk
7497     SUBEXIT
7500   END IF
7503   !
7506   OFF KBD
7509   OFF KEY
7512   OFF KNOB
7515   DEG
7518 Redo:Bmin=Min_barrel+3+4*(Filament(3)+Filament(4))0   ! Lower barrel-limit
7521   Bmax=Max_barrel-7-4*(Filament(3))0 OR Filament(4))0) ! Upper barrel-limit
7524   MAT Daly_ok= (0)
7527   Daly_ok(L)=Mu
7530   Barrelscans=0
```

```
7533   Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,1,Coarsemag(L<>0),Foc(*))
7536 Brun:OUTPUT 8;Mm$(2+Mu,1)
7539   Enter_beam(0,Mv0,1,1,1)
7542   IF Mv0<15 AND Daly AND (Mu=0) THEN  ! small beam- use Daly
7545     Mtest
7548     GOTO Brun
7551   END IF
7554   !
7557   IF NOT Auto AND Mu*(Mv0<.05) OR (NOT Mu)*(Mv0<2) THEN
7560     PRINT USING "10/,6X,K,3/";FNH$("CAN'T CENTER BARREL ACCURATELY UNLESS A SIGNIFICANT BEAM IS PRESENT")
7563     PRINT USING "16X,K,10/";"press  k9  to escape, or  k0  to try anyway."
7566     Clunk
7569     ON KEY 0 LABEL "    TRY" GOTO 7599
7572     ON KEY 9 LABEL "    ESCAPE" GOTO 7581
7575     GOTO 7575
7578     !
7581     OUTPUT KBD;CHR$(255)&CHR$(75);
7584     Subflag=4
7587     OUTPUT 8;Mm$(Mu,I_t0) ! restore initial integration time
7590     SUBEXIT
7593   END IF
7596   !
7599   FOR I=0 TO 19
7602     ON KEY I LABEL "" CALL Clunk
7605   NEXT I
7608   ON KEY 9 LABEL "    ESCAPE" GOTO Exit_barrel
7611   !
7614   I_t=1+Mu*(Mv0>1)+(NOT Mu)*(Mv0>10)
7617   OUTPUT 8;Mm$(2+Mu,I_t)
7620   DISP "ORIGINAL =";Barrel_pos0
7623   OUTPUT KBD;CHR$(255)&CHR$(75);
7626   IF NOT Barrelscans THEN
7629     Ymax=1.4*Mv0
7632     IF Ymax>1.E+4 THEN Ymax=1.E+4
7635     Axes(0,100,25,100,(Bmin),Bmax+1,0,Ymax,"BARREL POSITION","mV "&VAL$(Magnet(L,1)),1,Mu,0)
7638     LORG 5
7641     MOVE (Bmin+Bmax)/2,.1*Mv0
7644     CSIZE 4.5
7647     IF NOT Barrelscans THEN LABEL "BARREL-FOCUS"
7650     MOVE Barrel_pos0,1.76*Mv0
7653     LINE TYPE 4
7656     DRAW Barrel_pos0,.2*Mv0
7659     MOVE Barrel_pos0,.55*Mv0
7662     LINE TYPE 1
7665     LORG 4
7668     CSIZE 3
7671     LDIR 90
7674     LABEL "(ORIGINAL)"
7677     LDIR 0
7680   END IF
7683    !
7686 Bgo:CALL Brl(Bmin,Flag,1)
7689   GOSUB Check
7692   Barrel_position=Bmin
7695   CALL Brl(Barrel_position,Flag,.3)
7698   GOSUB Check
7701   DISP
7704   IF FNFiltest(Nfils,Flag)=0 AND NOT Refind THEN  ! lost contact at lower barrel-range
7707     Error_message(Prtr(*),1+2*Full_auto,"UNABLE TO KEEP FILAMENT-CONTACT DURING BARREL-ROTATION")
7710     IF F_bcheck=0 THEN
```

```
7713      CALL Flag(Flg(*),2+Mu)
7716      Flag=Flg(1)
7719      F_bcheck=1
7722      GOTO 7704
7725    END IF
7728    GOSUB Refind
7731    Refind=1
7734    GOTO Redo
7737  END IF
7740  Started=0
7743  WAIT 1
7746  FOR I=1 TO Bmax-Bmin+1   !  scan barrel
7749    GOSUB Barrelscan
7752  NEXT I
7755  !
7758  LOOP
7761    CALL Brl(Barrel_position,Flag,0)
7764    Ok=(FNFiltest(Nfils,Flag)) ! lost contact at upper barrel-range?
7767  EXIT IF Ok OR F_acheck
7770    Error_message(Prtr(*),1+2*Full_auto,"UNABLE TO KEEP FILAMENT-CONTACT DURING BARREL-ROTATION")
7773    F_acheck=1
7776    WAIT .2
7779  END LOOP
7782  IF NOT Ok THEN
7785    GOSUB Refind
7788    Refind=2
7791    GOTO Redo
7794  END IF
7797  Max_beam=0
7800  FOR J=1 TO Bmax-Bmin+1
7803    IF Max_beam<Barrel_pk(J) THEN
7806      Barrel_position=Bmin+J
7809      Max_beam=Barrel_pk(J)
7812    END IF
7815  NEXT J
7818  PRINT
7821  MOVE Barrel_position-.06,.5*Ymax
7824  RECTANGLE .12,.85*(Max_beam-.5*Ymax),FILL ! draw a pointer to new max
7827  !
7830  CALL Brl(Bmin,Flag,1)
7833  GOSUB Check
7836  CALL Brl(Barrel_position,Flag,1)
7839  OUTPUT 8;Mm$(Mu+2,1)
7842  Barrelscans=1+Barrelscans
7845  WAIT .5
7848  GOSUB Check
7851  WAIT .4
7854  Enter_beam(0,Remax,1,I_t,1)
7857  IF (Remax<.85*Max_beam) AND (Barrelscans<2) THEN Bgo  ! Re-scan if didn't recover at least 85% of the original beam-intensity
7860  IF Barrelscans<2 AND ((Bmax-Barrel_position<3) OR (Barrel_position-Bmin<3)) THEN
7863    Barrel_pos0=Barrel_position
7866    CALL Brl(Min_barrel+2,Flag,.5)
7869    GOTO Bgo   ! reoptimize barrel if position is within 2 units of losing contact.
7872  END IF
7875  Barrel_pos0=Barrel_position
7878  OUTPUT 8;Mm$(Mu,I_t0) ! restore integration time of calling environment
7881  DISP
7884  Daly_ok(L)=Mu
7887  Whoop
7890  SUBEXIT
```

```
7893    !
7896 Barrelscan:Barrel_position=Barrel_position+1
7899    CALL Brl(Barrel_position,-1,.2*(I_t=2),0)
7902    Enter_beam(0,P2(2),1,I_t,Pr)
7905    IF Pr=2 THEN Brun
7908    IF Pr=3 THEN Bgo
7911    IF P2(2)<=0 THEN CALL Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L<>0),Foc(*))
7914    Barrel_pk(I)=P2(2)
7917    P2(1)=Barrel_position !+.5
7920    IF Started THEN CALL Thickpen(P1(*),P2(*),Mv0/40,(2-Barrelscans)/2)
7923    Started=1
7926    MAT P1= P2
7929    RETURN
7932    !
7935 Exit_barrel:Barrel_position=Barrel_pos0
7938    Subflag=4
7941    Broop
7944    GOTO 7830
7947    !
7950 Check:! check for runaway barrel. If found, turn off fils, rotate sample back into position again, take up fils to previous cur
rents, re-center barrel
7953    FOR G=0 TO 1
7956      WAIT .4
7959      IF I_t=1 THEN OUTPUT 8;"$OMW9?=8"
7962      IF I_t=2 THEN OUTPUT 8;"$OMW9?68"
7965      OUTPUT 8;"$IMW"
7968      ENTER 8;P    ! P=1 if the barrel is in motion.
7971      IF P=0 THEN  ! barrel stopped moving- everything ok
7974        OUTPUT 8;Mm$(Mu+2,I_t)
7977        RETURN
7980      ELSE
7983        BEEP 100,.05
7986      END IF
7989    NEXT G
7992    !
7995    IF Find_flag AND Full_auto THEN ! only permit 1 attempt at recalibration
7998 Rats:Subflag=1    ! Rats.  Can't keep contact and/or non-runaway barrel.
8001      Error_message(Prtr(*),3,"UNABLE TO KEEP FILAMENT-CONTACT DURING BARREL-ROTATION")
8004      Zero_fils(Filament(*))
8007      SUBEXIT
8010    END IF
8013    !
8016    MAT F1= Filament
8019    Zero_fils(Filament(*))
8022    Superclunk
8025    ALPHA ON
8028    GRAPHICS OFF
8031    FOR I=1 TO 1+Auto
8034      PRINTER IS Prtr(I)
8037      PRINT USING "3/,2(9X,60A,/),9X,K,9X,60A";RPT$("*",60),RPT$("*",22)&" RUNAWAY BARREL "&RPT$("*",22),RPT$("*",60)
8040      PRINT USING "3/"
8043      IF I=1 THEN
8046        PRINT "Sorry- something has caused the barrel to rotate on its own."
8049        PRINT USING "2(/,K),/";"Keep calm.  The computer will rotate the sample back into position,","and restore the filament-cu
rrents  (honest!)."
8052      END IF
8055    NEXT I
8058    PRINTER IS CRT
8061    Find(0,Sample,Nfils,F_subflag,Filament(*),0)
8064    IF F_subflag THEN 8061
```

```
8067   Find_flag=1+Find_flag
8070   GOSUB Restore_fils
8073   GOTO Redo
8076    !
8079 Refind:IF Find_flag AND Full_auto THEN Rats
8082   !
8085   MAT F1= Filament
8088   GRAPHICS OFF
8091   Clunk
8094   IF NOT Auto AND (Max_barrel-Min_barrel>15) THEN
8097     PRINT "PRESS "&FNH$("k0")&" FOR AN AUTOMATIC CONTACT-RECALIBRATION,"
8100     PRINT USING "/,K,2/,K";"PRESS "&FNH$("k4")&" TO JUST NARROW THE SCAN-RANGE BY 5 UNITS."
8103     OFF KEY
8106     ON KEY 0 LABEL "RECALIBRATE" GOTO 8139
8109     ON KEY 4 LABEL "SIMPLE ADJUST" GOTO 8118
8112     GOTO 8112
8115     !
8118     IF Lower THEN Min_barrel=Min_barrel+5
8121     IF NOT Lower THEN Max_barrel=Max_barrel-5
8124     CALL Brl((Max_barrel+Min_barrel)/2,-1,.5)
8127     OFF KEY
8130     RETURN
8133   END IF
8136   !
8139   PRINT USING "18/,K,2/";FNH$("RECALIBRATING CONTACT POSITIONS FOR THIS SAMPLE - PLEASE DON'T INTERVENE.")
8142   PRINT "(filament-contacts were lost during the barrel-scan)"
8145   Find(1,Sample,Nfils,F_subflag,Filament(*),0)
8148   IF F_subflag THEN 8061
8151   Find_flag=1+Find_flag
8154   OFF KEY
8157   GOSUB Restore_fils
8160   RETURN
8163    !
8166 Restore_fils:FOR I=1 TO 4     ! take up fil-currents to initial values
8169     IF (I=1 AND Nfils=1) OR (I=2 AND Nfils=2) THEN
8172     ! take up sample-filament rapidly to 85% of orig. curr, then slowly to orig. curr
8175       CALL Filament(200,.85*F1(I),.5,(I))
8178       CALL Filament(10,F1(I),.5,(I))
8181     ELSE
8184       IF F1(I) THEN CALL Filament(200,F1(I),.5,(I))
8187     END IF
8190   NEXT I
8193   MAT Filament= F1
8196   IF Nfils=0 THEN Nfils=1
8199   IF (Filament(3)=0) AND (Filament(4)=0) THEN CALL Wait(TIMEDATE,15,Filament(Nfils),Magnet(L,1),Auto,Full_auto)
8202   RETURN
8205 SUBEND  ! ------------------------------------------------------------
8208 !
8211 !
8214 Filament:SUB Filament(Rate,Target,Pcheck_amps,INTEGER Filnumber)
8217  ! change current of specified filament to target-value at specified speed
8220   OPTION BASE 1
8223   COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
8226   COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
8229   COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
8232   COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
8235   COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
8238   COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
8241   DIM Flag(2)
8244 !
```

```
8247   IF NOT Full_auto THEN OUTPUT KBD;Clear$;
8250   Subflag=0
8253   OFF KNOB
8256   IF Pcheck_amps=0 THEN Pcheck_amps=1 ! just in case 0 accidentally passed
8259   Target=DROUND(Target,4) ! because of binary-decimal roundoff errors
8262   FOR I=Auto*Full_auto*2 TO 19
8265     ON KEY I LABEL "" CALL Clunk
8268   NEXT I
8271   ON KEY 9 LABEL "    ESCAPE" GOTO Exit
8274   ON KEY 3 LABEL " DOUBLE RATE" GOTO Double_rate
8277   ON KEY 8 LABEL " HALVE  RATE" GOTO Halve_rate
8280    !
8283   Max_pressure=1.E-6  ! Maximum permissible source-pressure
8286   IF Rate<.04 THEN Rate=.04  ! (otherwise, get waits of >30 seconds)
8289   F=Filament(Filnumber)
8292   Start=F
8295   Pcheck_start=F
8298   IF Target>1 AND Target-Start>.25 THEN GOSUB Check_pressure
8301   Preheat=(Filnumber>2)
8304   IF Target>Start OR Rate<1 THEN Mu=0
8307   OUTPUT 8;Mm$(Mu,1)
8310 !
8313 Takeup_current:Increment=.001*10^((Rate>10)+(Rate>1000))     ! step increment (amps)
8316   W=10^((Rate>10)+(Rate>1000))/Rate  ! step wait (seconds)
8319   Target=INT(Target*1000)/1000
8322   IF Target<Start THEN No_press_check=1  ! don't check pressure if reducing current
8325   Met_target=0
8328   !
8331   LOOP
8334     F=F+Increment*SGN(Target-F)
8337     IF NOT Flagchecked AND (F>.2) THEN
8340       FOR P=0 TO 1
8343         FOR Fct=1 TO 2! Put 0.2 amps at least thru all fils for flag-check
8346           IF Filament(Fct)=0 THEN OUTPUT 8 USING "4A,4Z";Ff$(P,Fct),200
8349         NEXT Fct
8352       NEXT P
8355       CALL Flag(Flag(*),0) ! Check for valid filament contacts
8358       FOR P=0 TO 1
8361         FOR Fct=1 TO 2! rezero fils
8364           IF Filament(Fct)=0 THEN OUTPUT 8;Ff$(P,Fct),0
8367         NEXT Fct
8370       NEXT P
8373       OUTPUT 8;Mm$(0,1)
8376       Flagchecked=1
8379       IF (Filnumber=1)*((Flag(1)=1)+(Flag(1)=3)) OR (Filnumber=2)*(Flag(1)>1) THEN 8409
8382       IF (Filnumber=3)*((Flag(2)=1)+(Flag(2)=3)) OR (Filnumber=4)*(Flag(2)>1) THEN 8409
8385       Subflag=1
8388       PRINT USING "2/,K";"******** NO FILAMENT-CONTACT - CORRECT PROBLEM & TRY AGAIN ********"
8391       PRINT USING "K,2/";"(are center and/or side filament switches in the on position?)"
8394       Clunk
8397       WAIT 4
8400       SUBEXIT
8403     END IF
8406    !
8409     OUTPUT 8 USING "4A,4Z";Ff$(Preheat,1+(Filnumber=2 OR Filnumber=4)),FNF(F)
8412     Filament(Filnumber)=F
8415     Met_target=ABS(F-Target)<(Increment-.0001)
8418   EXIT IF Met_target
8421   !
8424     DISP Ci$;Magnet(L,1);Cn$;TAB(14);Fil$(Filnumber);"=";F;"AMPS";"   -TARGET=";Target;"AMPS,   RATE=";Rate;"mA/SEC."
```

```
8427     IF W>.8 THEN ! so can exit with k9 key immediately
8430       FOR I=1 TO W/.2
8433         WAIT .2
8436       NEXT I
8439     ELSE
8442       WAIT W
8445     END IF
8448 ! check source pressure if current> 1 amp every Pcheck_amps
8451     IF DROUND((F-Pcheck_start)/Pcheck_amps,4)=INT(DROUND((F-Start)/Pcheck_amps,4)) AND F>1 AND NOT No_press_check THEN
8454   !
8457       LOOP
8460         CALL Pressure(0,Mm$(*),1,Source_pressure)
8463       EXIT IF Source_pressure<Max_pressure
8466     !
8469         Clunk
8472         PRINT "WAITING FOR SOURCE-PRESSURE TO DECREASE TO (";1.E+7*Max_pressure;"E-07..."
8475         Wait(TIMEDATE,20,Filament(Nfils),0,Auto,Full_auto)
8478       END LOOP
8481     END IF
8484   END LOOP
8487       !
8490   IF Met_target THEN
8493     Filament(filnumber)=Target
8496     OUTPUT 8 USING "4A,4Z";Ff$(Preheat,1+(Filnumber=2 OR Filnumber=4)),FNF(Target)
8499   END IF
8502     !
8505 Exit:IF Target>Start OR Rate<1 THEN MAT Daly_ok= (0)
8508 !   OUTPUT KBD;Clear$;
8511   SUBEXIT
8514 !
8517 No_press_check:No_press_check=1
8520   BEEP 440,.1
8523   PRINT "(no pressure-checks)"
8526   ON KEY 4 LABEL "CHECK PRESSURE" GOSUB Check_pressure
8529   RETURN
8532                     !
8535 Check_pressure:No_press_check=0
8538   IF F>Start THEN BEEP 440,.1
8541   PRINT "(pressure-check every ";Pcheck_amps;" amperes)"
8544   ON KEY 4 LABEL "NO PRESS-CHECK" GOSUB No_press_check
8547   RETURN
8550 !
8553 Double_rate:Rate=Rate*2
8556   Pcheck_start=F
8559   BEEP 1000,.08
8562   GOTO Takeup_current
8565                 !
8568 Halve_rate:Rate=Rate/2
8571   BEEP 220,.08
8574   GOTO Takeup_current
8577 SUBEND  ! ------------------------------------------------------------
8580 !
8583 !
8586 Mtest:SUB Mtest    ! Check that peak is small enough for Daly
8589   OPTION BASE 1
8592   COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
8595   COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
8598   COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
8601   COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
8604   COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
```

```
8607   OFF KNOB
8610   IF Daly=0 THEN
8613     IF Auto THEN SUBEXIT
8616     Clunk
8619     PRINT TABXY(1,18);FNH$("DALY DISABLED")&"   (press [CONTROL] D during BMC to enable)"
8622     SUBEXIT
8625   END IF
8628   IF Daly_ok(L) THEN Turn_on_daly
8631    !
8634   PRINTER IS CRT
8637   I_t0=I_t
8640   I_t=2
8643   OUTPUT 8;Mm$(0,I_t)  ! Faraday cup, .2-sec integration
8646   FOR K=1 TO 7       ! scan magnet from -3/8 to +3/8 isotope
8649     GOSUB Check_peak
8652     IF Mv>50 THEN Peak_too_large
8655     IF Mv>Mv_max OR K=1 THEN
8658       Mv_max=Mv
8661       Max_mag=Magnet_value
8664     END IF
8667   NEXT K
8670    !
8673 Turn_on_daly:PRINT USING "/,K,K,8X,K,/";Magnet(L,1)," PEAK OK FOR DALY",FNH$("DALY ON")
8676   IF NOT Auto THEN BEEP 1000,.1
8679   IF Daly_ok(L)=0 AND Mv_max>4 THEN  ! check to see if Daly functioning OK
8682     OUTPUT 8 USING "4A,4Z";"$OFJ",Max_mag
8685     Mu=1
8688     Enter_beam(X,Mv,L,2,1,.4,0)
8691     IF Mv<Mv_max/2.5 THEN  ! something wrong
8694       PRINT USING "5/,13X,K,4/,K";" DALY-DETECTOR RESPONSE IS INADEQUATE ","PLEASE CHECK THAT THE MULTIPLIER SUPPLY AND FA
3 SWITCHES ARE  ON ,"
8697       PRINT USING "/,K,2/";"AND THAT THEIR KNOBS ARE AT THE MARKED POSITIONS."
8700       PRINT USING "K,2/,K";"MAKE SURE THAT THE BRANDENBURG IS  ON ,","AND THEN PRESS THE BRANDENBURG  RESET  BUTTON."
8703       Superclunk
8706       IF Auto THEN
8709         WAIT 10
8712         Daly=0
8715         Mu=0
8718       ELSE
8721         PRINT USING "4/,K";"-- Press CONTINUE to return to the BMC --"
8724         Mu=0
8727         PAUSE
8730       END IF
8733       GOTO 8778
8736     END IF
8739   END IF
8742   Mu=1
8745   Daly_ok(L)=1
8748   GOTO 8793
8751    !
8754 Check_peak:Magnet_value=INT(Magnet(L,2)+Peak_inter*(K-4)/8)
8757   OUTPUT 8 USING "4A,4Z";"$OFJ",Magnet_value
8760   Enter_beam(X,Mv,L,2,1,.2,0)
8763   DISP "CHECKING";Magnet(L,1);"INTENSITY FOR DALY",Magnet_value;X
8766   RETURN
8769              !
8772 Peak_too_large:PRINT TABXY(1,18);Magnet(L,1);"PEAK TOO LARGE FOR DALY"
8775   !
8778   IF NOT Auto THEN CALL Clunk
8781   Daly_ok(L)=0   !  this peak must be checked again if Daly is requested
```

```
8784   Mu=0
8787    !
8790   I_t=I_t0
8793   OUTPUT 8 USING "4A,4Z,8A";"$OFJ",Magnet(L,2),Mm$(Mu,I_t)
8796   DISP
8799 SUBEND  ! ----------------------------------------------------------
8802 !
8805 !
8808 Correct:SUB Correct(Ff$(*),Mm$(*),REAL Filament(*),INTEGER Magnet,Mu,I_t,Coarsemag,Foc(*))
8811  ! Change/restore magnet, focus, filament, & collector values
8814   Z$="123456789"
8817   OUTPUT 8 USING "2(4A,4Z),8A";"$OFJ",Magnet,"$OFK",Coarsemag  ! magnet values
8820   OUTPUT 8 USING "2(4A,4Z),8A";Ff$(0,1),FNF(Filament(1)),Ff$(0,2),FNF(Filament(2))  ! sample-filament values
8823   OUTPUT 8 USING "2(4A,4Z),8A";Ff$(1,1),FNF(Filament(3)),Ff$(1,2),FNF(Filament(4))  ! preheat-filament values
8826   OUTPUT 8:Mm$(Mu,I_t)  ! collector-type & integration-time
8829   FOR I=1 TO 8  !  focus values
8832     OUTPUT 8 USING "4A,4Z";"$OF"&Z$[10-I,10-I],Foc(I)
8835   NEXT I
8838 SUBEND  ! ----------------------------------------------------------
8841 !
8844 Pressure:SUB Pressure(INTEGER Mu,Mm$(*),I_t,REAL Source_press,OPTIONAL Tube_pressure)
8847  ! Query source-pressure and (optional) tube-pressure
8850   OFF KEY
8853   OFF KNOB
8856   ON KEY 9 LABEL "    ESCAPE" GOTO 8919
8859   V$=Mm$(Mu,1)[8]
8862   PRINTER IS CRT
8865   PRINT "SOURCE PRESSURE =";
8868   OUTPUT 8;"$OMW04:"&V$
8871   FOR I=1 TO 10
8874     WAIT .2
8877   NEXT I
8880   OUTPUT 8;"$IDV"
8883   ENTER 8;P0     !           zero-value
8886   OUTPUT 8;"$OMW00:"&V$
8889   GOSUB Raw_press
8892   Source_press=DROUND(1.E-8*10^((P-P0-2950)/1020),3)
8895   PRINT Source_press;
8898   IF NPAR=5 THEN
8901     PRINT TAB(40);"TUBE PRESSURE =";
8904     OUTPUT 8;"$OMW02:"&V$
8907     GOSUB Raw_press
8910     Tube_pressure=DROUND(10^(7.185E-6*(P-P0)-8.969),3)
8913     PRINT Tube_pressure;
8916   END IF
8919   PRINT
8922   OUTPUT 8;Mm$(Mu,I_t)
8925   SUBEXIT
8928   !
8931 Raw_press:FOR I=1 TO 5
8934     WAIT .2
8937   NEXT I
8940   OUTPUT 8;"$IDV"
8943   ENTER 8;P
8946   IF P<P0 THEN P=P0
8949   RETURN
8952 SUBEND  ! ----------------------------------------------------------
8955 !
8958 !
8961 Contact_test:SUB Contact_test(Filament(*),Time0,Escape,Sample_name$(*),Run_name$(*),Candump,INTEGER Estbar(*),Sample,Nfils)
```

```
8964  !
8967  ! locate barrel-positions with valid filament-contacts for each sample
8970  !
8973   OPTION BASE 1
8976   COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
8979   COM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel
8982   COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
8985   DIM Flag(2)
8988   INTEGER Contact(200)
8991   Escape=1
8994   OFF KEY
8997   OFF KNOB
9000   ON KEY 0 LABEL "   START" GOTO 9015
9003   ON KEY 9 LABEL "    ESCAPE" GOTO 9405
9006   PRINT USING "8/,K,/,K,/";"PRESS   k0   TO CHECK FILAMENT-CONTACTS FOR ALL OF THE SAMPLES IN THE BARREL","(turns filaments
OFF),"
9009   PRINT "PRESS   k9   TO ESCAPE  (return to BMC)."
9012   GOTO 9012
9015   OFF KEY
9018   OFF KBD
9021     !
9024   PRINT USING "18/,17X,K,8/";"PLEASE WAIT WHILE BARREL IS BEING RESET"
9027   Escape=0
9030   Resbar
9033   MAT Estbar= (0)
9036   OUTPUT 8;"$OM00200$OM10200$OP00200$OP10200" ! put .2 amps thru all fils
9039   Axes(0,100,25,100,0,17,-40,40,"BARREL-NUMBER","BARREL UNITS",1,-1,0)
9042   GRID 1,0,0,0,1,0,0
9045   AXES 0,2,0,0,0,5,3
9048   AXES 0,2,17,80,0,5,3
9051   LINE TYPE 4
9054   GRID 0,2,0,0,0,5,3
9057   LINE TYPE 1
9060  !
9063   ON KEY 9 LABEL "   ESCAPE" GOTO 9399
9066   Barnum=1
9069   OUTPUT 8;Mm$(2,1) ! enable barrel-motor
9072   CSIZE 3.2
9075   LORG 5
9078   FOR Barpos=1 TO 16 ! check contacts over possible range for each barrel#,
9081                      ! and graphically indicate regions of contact for both
9084                      ! sample and preheat filaments.
9087     Barnum=FNBarnum(Barpos)
9090     MAT Contact= (0)
9093     Est=81+187*(Barpos-1) ! estimated position
9096     Start=Est*(Barnum=1)+70*(Barnum>1)
9099     CALL Brl((Est-Start)*(Est>Start),0,2,0) ! rotate to est. pos. minus 70
9102     FOR J=Est-Start TO Est+60 ! define possible range as -start to +60 units
9105       P=J-Est+Start+1
9108       CALL Brl((J),-1,.10,0)
9111       DISP "SAMPLE# "&FNH$(VAL$(Barnum));TAB(22);"Barrel-position: ";J
9114       CALL Flag(Flag(*),2,1)
9117       Contact(P)=Flag(1)
9120        !
9123       FOR F=1 TO 2
9126         IF Flag(F) THEN
9129           C=(Flag(F)=3)+.5*(Flag(F)=1)   ! box-pattern is solid for triple,
9132           AREA COLOR C,C,C ! gray for single-fil !         gray for single,
9135           MOVE Barnum+.35*(F-2),J-Est-1/2       !     blank for sides-only
9138           ! draw rectangle whose fill-pattern indicates which filaments are
```

```
9141          ! contacted (to left of barrel# for sample, right for preheat).
9144          RECTANGLE .35,1,FILL,EDGE
9147        END IF
9150      NEXT F
9153      !
9156    NEXT J
9159      !
9162    DATA 0,0,0,0,0,0,0,0
9165    RESTORE 9162
9168    READ S,T,S1,T1,S2,T2,Z
9171    FOR P=1 TO 200          ! calculate best default contact-positions
9174      C=Contact(P)
9177      S=S+((C=1) OR (C=3))   ! #successive single-fil contacts
9180      T=T+(C=3)    !               "      triple  "     "
9183      IF (S)10 OR T)10) THEN  ! count # of successive nocontacts
9186        Z=Z+(C=0 OR C=2)
9189        IF Z)10 THEN 9234
9192      END IF
9195      IF C=1 AND S1=0 THEN S1=P                 ! 1st single-fil contact
9198      IF C=3 AND T1=0 THEN T1=P                 ! "   triple  "    "
9201      IF C=0 AND S2=0 AND S)=10 THEN S2=P-1      ! last single-fil contact
9204      IF C(3 AND T2=0 AND T)=10 THEN T2=P-1      ! "   triple  "    "
9207      IF ((C=0) OR (C=2)) AND S1 AND S(10 THEN   ! don't count contact widths of (10
9210        S=0
9213        S1=0
9216      END IF
9219      IF C(3 AND T2 AND S(10 THEN               ! "    "     "     "
9222        T1=0
9225        T=0
9228      END IF
9231    NEXT P
9234    IF S)=10 THEN    ! safest estimate of single-fil contact-position
9237      Estbar(Barnum,1)=Est-Start+(S2+S1)/2
9240      Estbar(Barnum,2)=1
9243    END IF
9246    IF T)=10 THEN    ! ditto, triple-fil.
9249      Estbar(Barnum,2)=Est-Start+(T1+T2)/2
9252      Estbar(Barnum,2)=2
9255    END IF
9258  NEXT Barpos
9261 Exit_test:Zero_fils(Filament(*))
9264  OFF KEY
9267  CALL Brl(3080,-1,1,0)
9270  FOR Barpos=1 TO 16 ! find first sample in the barrel to rotate to when done
9273    IF Estbar(FNBarnum(Barpos),1) THEN 9285
9276  NEXT Barpos
9279  Sample=1 ! default
9282  !
9285  Sample=FNBarnum(Barpos)
9288  OUTPUT KBD;Clear$;
9291  Nfils=Estbar(Sample,2)
9294  IF Nfils=0 THEN Nfils=1
9297  IF Candump AND NOT Escape THEN
9300    DISP CHR$(130)&"Dumping Contact-Test Graphics..."&CHR$(128)
9303    DUMP GRAPHICS
9306  END IF
9309  DISP FNH$("DO YOU WANT TO ERASE THE PREVIOUSLY-DEFINED SAMPLE NAMES NOW?  (Y/N)");
9312  INPUT Temp$
9315  IF FNYes(Temp$)=1 THEN
9318    MAT Sample_name$= ("")
```

```
9321    MAT Run_name$= ("")
9324  END IF
9327  !
9330  OFF KEY
9333  LOOP
9336    LOOP
9339      INPUT "WHAT BARREL# DO YOU WANT ROTATED INTO RUNNING POSITION?",Sample
9342    EXIT IF Sample)=1 AND Sample(=16
9345      PRINT USING "/,10X,K,/";FNH$(" **** BARREL-NUMBERS ONLY GO UP TO 16 **** ")
9348      Clunk
9351    END LOOP
9354  EXIT IF Estbar(Sample,1) ! did the contact-test find a sample here?
9357    Clunk
9360    GRAPHICS OFF
9363    PRINT TABXY(1,18);FNH$("NO SAMPLE AT BARREL# "&VAL$(Sample)&" --  PLEASE RE-ENTER BARREL#...")
9366  END LOOP
9369  !
9372  DISP "PRESS CONTINUE TO RESET BARREL, ROTATE BARREL#";Sample;"INTO POSITION, & ESCAPE TO BMC"
9375  PAUSE
9378  GRAPHICS OFF
9381  OUTPUT KBD;Clear$;
9384  OFF KEY
9387  OFF KBD
9390  Resbar
9393  Find(0,Sample,Nfils,0,Filament(x),Time0)
9396  SUBEXIT
9399  Escape=1
9402  GOTO Exit_test
9405 SUBEND ! ----------------------------------------------------------------
9408 !
9411 !
9414 Average:SUB Average(Auto,Normal(*),Candump,Escape,INTEGER Run_number,Prtr(*))
9417  OPTION BASE 1
9420  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
9423  Maxblox=80 ! maximum # of blocks of data that can be brought into memory
9426  DIM Date$[12],Samplename$[50],Ratio$(80,7)[7],Block_ratio$(7)[7],Aver_ratio$[7],Ra$(20)[7],J$[100],Input$[160],Delta$(7)[10],
Sigma$(7)[10],Nfrag$[10]
9429  REAL Aver(80,7),Ok_ratio1(80),Block_average(7)
9432  REAL Acc(80,7),Block_acc(7),Ok_acc(80),Pk,Rejected(80),Blocktime(80),Ok_time(80)
9435  INTEGER Sample,Ref,Block,Data_collector,N_isotopes(80),Nblox(80),N
9438  Escape=0
9441  Run=Run_number
9444  Rnum=Run
9447  First_average=1
9450  OUTPUT KBD;Clear$;
9453  GRAPHICS OFF
9456  OFF KNOB
9459  PRINT USING "16X,K,/";FNUn$(" WEIGHTED AVERAGES OF RATIOS: ")
9462  IF NOT Auto THEN
9465    IF First_average THEN
9468      PRINTER IS CRT
9471      PRINT USING "8/,22X,K";"  WHICH RUN#?  "
9474      PRINT TABXY(1,14);"enter 0 to escape,  a negative run# (e.g. -"&VAL$(Run_number)&") for CRT-display only,"
9477      PRINT TABXY(1,16);"press CONTINUE for current run ("&VAL$(Run_number)&"),"
9480      PRINT TABXY(1,18);"enter 100 to use file numbers rather than Run numbers)"
9483    ELSE
9486      DISP "WHICH RUN#? (0 to escape, CONT for run# "&VAL$(Rnum)&"), neg. run# for CRT-display only";
9489    END IF
9492    INPUT Rnum
9495    Files=0
```

```
9498    IF Rnum=100 THEN
9501      INPUT "ENTER FIRST, LAST FILE #S FOR DATA:",First_record,Last_record
9504      Files=1
9507    ELSE
9510      IF Rnum=0 THEN
9513        Escape=1
9516        SUBEXIT
9519      END IF
9522      Run_number=ABS(Rnum)
9525    END IF
9528  END IF
9531  Print=(Rnum>0) !Provide hard-copy printout IF Print=1 (i.e. negative run#)
9534  PRINTER IS CRT
9537  GRAPHICS OFF
9540  IF Run_number<>Lastrun OR NOT Got OR Files THEN
9543    IF NOT Files THEN
9546      Lastrun=Run_number
9549      ON ERROR GOTO Faildir
9552      ASSIGN @Path1 TO "RESDIR:INTERNAL,4,1"
9555      ENTER @Path1,Run_number;Sample,Date$,Samplename$,First_record,Last_record
9558      PRINT USING "18/,K,/,8A,2D,10X,5A,2D,10X,K,2/,K,3X,#";"SAMPLE: "&Samplename$,"BARREL# ",Sample,"RUN# ",Run_number,Date$,"
ISOTOPE RATIOS:"
9561    END IF
9564    ON ERROR GOTO Faildat
9567    DISP "Getting data from disk.  Please wait..."
9570    ASSIGN @Path1 TO "RESULT:INTERNAL,4,1"
9573    FOR K=First_record TO Last_record+500*(First_record>Last_record)!get data from disk for all blocks of run
9576      I=K-500*(K>500)
9579      ENTER @Path1,I;Nfrag$,Block_ratio$(*),Block_average(*),Sigma$(*),Block_acc(*),Delta$(*),Pk,N,Data_collector,Block,Time,Fi
lcurr
9582      Aver_ratio$=""
9585      Got=1+Got! at least 1 block retrieved
9588      N_isotopes(Block)=N
9591      Blocktime(Block)=Time
9594      FOR J=1 TO N-1! put data for all blocks of run into arrays
9597        Aver(Block,J)=Block_average(J)
9600        Acc(Block,J)=Block_acc(J)
9603        Ratio$(Block,J)=Block_ratio$(J)
9606      NEXT J
9609      IF Block=Maxblox THEN PRINT USING "2/,3(K),2/";"*** CAN ONLY TAKE UP TO ",Maxblox," BLOCKS ***"
9612      IF Block=Maxblox THEN 9618
9615    NEXT K
9618    OFF ERROR
9621    R=0
9624    MAT Ra$= ("")
9627    FOR I=1 TO Block! find out which ratios were taken during the run and store in the Ra$ array
9630      FOR J=1 TO N_isotopes(I)-1
9633        Oldratio=0
9636        FOR K=1 TO R
9639          IF Ratio$(I,J)=Ra$(K) THEN Oldratio=1
9642        NEXT K
9645        IF NOT Oldratio THEN
9648          R=1+R! Oldratio=0 if a new ratio
9651          Ra$(R)=Ratio$(I,J)
9654          IF R=20 THEN 9672 ! no more than 20 different ratios can be handled
9657        END IF
9660      NEXT J
9663    NEXT I
9666  END IF
9669  !
```

```
9672  IF R=1 AND NOT Auto THEN
9675    Aver_ratio$=Ra$(1)
9678    PRINT
9681    GOTO 9750
9684  END IF
9687  IF NOT Auto THEN
9690    FOR I=1 TO R
9693      PRINT USING "k,3X,#";Ra$(I)
9696    NEXT I
9699    PRINT USING "2/"
9702    IF Aver_ratio$="" THEN Aver_ratio$=Ra$(1)
9705    DISP "WHICH RATIO (xxx/xxx)?  (CONTINUE for "&Aver_ratio$[1,7]&")";
9708    INPUT Aver_ratio$[1,7]
9711    Aver_ratio$=TRIM$(UPC$(Aver_ratio$))
9714    Rct=1 !####????
9717    GOTO 9744
9720  END IF
9723   !
9726  FOR Rct=1 TO N_isotopes(Block)-1
9729    Aver_ratio$=Ratio$(Block,Rct)
9732    IF Normal(1) AND POS(Aver_ratio$,VAL$(Normal(1))) THEN 10005   ! if an automatic run, don't average the normalizing ratio
9735    IF (Ratio$(Block,N_isotopes(Block)-1)="SAM/SPK") AND (Aver_ratio$()"SAM/SPK") AND (NOT POS(Aver_ratio$,"*")) THEN 10005
9738       ! don't average un-normalized spiked-run ratios
9741   !
9744    PRINTER IS Prtr(1+Auto)
9747    IF Full_auto THEN PRINT USING "/,80A,/";RPT$("*",80)
9750    PRINT USING "17X,8A,5X,9A,9X,K,/";FNUn$("BLOCK#"),FNUn$(Aver_ratio$),FNUn$("SIGMAMEAN%")
9753    Found_ratios=0
9756    FOR I=1 TO Block    ! display all values for selected ratio
9759      FOR J=1 TO N_isotopes(I)-1
9762        IF Ratio$(I,J)=Aver_ratio$ THEN
9765          PRINT TAB(18);I;TAB(27);DROUND(Aver(I,J),6);TAB(44);DROUND(Acc(I,J),3)
9768          Found_ratios=1+Found_ratios
9771          GOTO 9780
9774        END IF
9777      NEXT J
9780    NEXT I
9783    IF NOT Found_ratios THEN
9786      PRINT USING "/,K,2/";FNH$(" THERE ARE NO "&Aver_ratio$&" RATIOS IN THIS RUN")
9789      ON 1+Auto GOTO 9462,10005
9792    END IF
9795    IF NOT Auto THEN PRINT USING "2/"
9798    N=0
9801    Input$=""
9804    IF NOT Auto THEN
9807      DISP "RATIOS TO BE REJECTED?  (e.g. 2,5,32) [ press "&FNUn$("CONT")&" for none]";
9810      LINPUT Input$
9813      Parse(Input$,Rejected(*),Ninputs)
9816      IF Print THEN PRINTER IS Prtr(2)
9819      N_rej_blocks=Ninputs
9822      IF N_rej_blocks>=Found_ratios THEN
9825        Clunk
9828        DISP FNH$("YOU CAN'T REJECT THAT MANY BLOCKS.")
9831        WAIT 3
9834        GOTO 9807
9837      END IF
9840      IF N_rej_blocks THEN
9843        PRINT "BLOCKS NOT INCLUDED:  ";
9846        FOR I=1 TO N_rej_blocks
9849          PRINT Rejected(I);
```

```
9852        NEXT I
9855        PRINT
9858      END IF
9861    END IF
9864    FOR I=1 TO Block
9867      FOR K=1 TO N_rej_blocks
9870        IF I=Rejected(K) THEN 9906
9873      NEXT K
9876      FOR J=1 TO N_isotopes(I)-1
9879        IF Ratio$(I,J)=Aver_ratio$ THEN
9882          N=1+N
9885          Ok_ratio1(N)=Aver(I,J)
9888          Ok_acc(N)=Acc(I,J)
9891          Ok_time(N)=Blocktime(I)
9894          Nblox(N)=I
9897          GOTO 9906
9900        END IF
9903      NEXT J
9906    NEXT I
9909    IF NOT Auto AND Print THEN
9912      FOR Y=0 TO Print! printout the ratios to be averaged
9915        PRINTER IS Prtr(Y+1)
9918        PRINT USING "80A,/,K,/,8A,2D,10X,5A,2D,/";RPT$("*",80),"SAMPLE: "&Samplename$,"BARREL# ",Sample,"RUN# ",Run_number
9921        PRINT USING "21X,9A,4X,9A,4X,K,/";FNUn$("BLOCK#"),FNUn$(Aver_ratio$),FNUn$("SIGMAMEAN%")
9924        FOR I=1 TO N
9927          PRINT TAB(22);Nblox(I);TAB(31);DROUND(Ok_ratio1(I),6);TAB(44);DROUND(Ok_acc(I),3)
9930        NEXT I
9933        PRINT
9936      NEXT Y
9939    END IF
9942  !
9945    IF N>1 THEN
9948      CALL Avgraf(Ok_ratio1(*),(N),Aver_ratio$,Ok_acc(*),Ok_time(*))
9951      Calcav(Ok_ratio1(*),Ok_time(*),Print,Aver_ratio$,Ok_acc(*),Candump,N,Prtr(*))
9954      ALPHA ON
9957      IF Print AND Candump THEN
9960        IF NOT Auto THEN
9963          DISP "DO YOU WANT TO DUMP THESE GRAPHICS TO THE PRINTER?"
9966          OFF KEY
9969          ON KEY 0 LABEL "DUMP GRAPHICS" GOTO 9981
9972          ON KEY 4 LABEL "   NO DUMP" GOTO 9993
9975          GOTO 9975
9978        END IF
9981        OFF KEY
9984        DISP FNB1$("DUMPING GRAPHICS...")
9987        DUMP GRAPHICS
9990      END IF
9993      First_average=0
9996      OFF KEY
9999      IF NOT Auto THEN 9462
10002   END IF
10005  NEXT Rct
10008  SUBEXIT
10011   !
10014 Faildir:PRINT USING "/,3(K),2/";Ci$&" CAN'T READ DATA-DIRECTORY FOR RUN# ",Run_number," "&Cn$
10017  ON 1+Auto GOTO 9462,10008
10020 Faildat:PRINT USING "/,5(K),2/";Ci$&" CAN'T READ DATA FOR FILE# ",K,", RUN# ",Run_number," "&Cn$
10023  GOTO 9615
10026 SUBEND ! ----------
10029!
```

```
10032!
10035 Endsave:! end of SAVEd part
10044 DEF FNBar_digits(Barslot) ! Rotational position of barrel
10045 ! MAR 22, 1985, 5:15 PM
10047 ! Barslot is the relative position in the barrel (1-16), not the
10050 ! barrel-number.
10053  Y=-103.3+185.72*Barslot+.1408*Barslot^2
10056  RETURN INT(Y)+(FRACT(Y)>.5)
10059 FNEND
10062 !
10065 !
10068 Pb_age:SUB Pb_age(R76)
10071  L8=1.55125E-4
10074  L5=9.8485E-4
10077  U=137.88
10080  IF R76>.0156 AND R76<1.9 THEN
10083    Trial_t=4500*(SGN(-(1/2)+(R76>L5/(L8*U)))+(R>.7))
10086    Change=1
10089    REPEAT
10092      F=L8*EXP(Trial_t*(L8-L5))/L5
10095      T=LOG(1+(EXP(Trial_t*L8)-1-F*(EXP(Trial_t*L5)-1))/(1/(U*R76)-F))/L5
10098      Change=ABS(T-Trial_t)
10101      Trial_t=T
10104    UNTIL Change<.01
10107  END IF
10110  IF T THEN
10113    PRINT TABXY(1,18);
10116    PRINT USING "K,D.4D,K";"207/206 AGE FOR ";R76;" = ";GROUND(T,5)
10119  END IF
10122 SUBEND
10125 !
10128 !
10131 Error_message:SUB Error_message(INTEGER Prtr(*),Printer,Message$)
10134   ! If Printer=1, then display on CRT only
10137   ! If Printer=2, then printout only
10140   ! If Printer=3 then do on both.
10143   !
10146  FOR P=1 TO 1+(Printer=3)
10149    PRINTER IS Prtr(P+(Printer=2))
10152    PRINT USING "4/,K,4/";FNN$("*****  "&Message$&" *****")
10155  NEXT P
10158  PRINTER IS CRT
10161 SUBEND
10164 !
10167 !
10170 Broop:SUB Broop   ! make a "droopy-beep" sound
10173  FOR I=500 TO 100 STEP -50
10176    BEEP I,.02
10179  NEXT I
10182 SUBEND
10185 !
10188 !
10191 Drift_adjust:SUB Drift_adjust(Type,Runclass$,Nuclide$(*),Normal0(*),Norm0,Na$,INTEGER Niso,Rf,Hv0,Magnet0(*),Coarsebin(*),Ref)
10194  ! Recalibrate appropriate elements for slight drift of magnet-settings
10197  OPTION BASE 1
10200  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
10203  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
10206  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
10209  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarserange
10212  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
```

```
10215  COM /Data1/ Peak(*),Pk,Interfere(*),Normal(*),TOO,INTEGER Data_iso(*),Data_collector,Inv,Bmc_out,Next_run,Spike
10218  INTEGER Hv
10221  OUTPUT KBD;CHR$(255)&CHR$(75);
10224  PRINT USING "8/,K,/";"This function will adjust the stored ELEMENT data for slight drifts in the best magnet-settings for the
 element."
10227  PRINT USING "K,/,K,#";"To do this, you must:  (1) have the high voltage set within 2 volts of the","default value for the ele
ment, and"
10230  PRINT USING "K,2/,K";"  (2) be on a peak of at least 10 mV.","If both of the above are true, then press the appropriate softk
ey to proceed."
10233  OFF KEY
10236  ON KEY 0 LABEL "   PROCEED" GOTO 10248
10239  ON KEY 9 LABEL "    ESCAPE" GOTO 10368
10242  GOTO 10242
10245  !
10248  OFF KEY
10251  OUTPUT KBD;CHR$(255)&CHR$(75);
10254  CALL Hv(Mm$(*),Hv1,P,Mu,I_t)          ! query accelerating voltage
10257  IF P THEN SUBEXIT
10260  IF ABS(Hv0-Hv1)>2 THEN
10263    PRINT USING "5/,6X,K,K";FNH$(" ***** HIGH-VOLTAGE MUST BE WITHIN 2 VOLTS OF "&VAL$(Hv0)&" ")
10266    Clunk
10269    WAIT 2
10272    SUBEXIT
10275  END IF
10278  Hv=Hv1
10281  WAIT 1
10284  Enter_beam(0,Mu,L,I_t,0)
10287  IF (Daly AND Mu<10) OR (NOT Daly AND Mu<20) THEN
10290    PRINT USING "3/,6X,K";FNH$(" **** MUST HAVE >10 mV (DALY)  OR >20 mV  (CUP) *** ")
10293    Clunk
10296    WAIT 2
10299    SUBEXIT
10302  END IF
10303  L0=L
10305  Center_peak(1,L)
10308  IF Subflag THEN
10311    PRINT USING "4/,6X,K";FNH$(" ***** NEED A CENTERABLE PEAK ******")
10314    Clunk
10317    WAIT 2
10318    SUBEXIT
10319  END IF
10320  !
10321  IF Filament(1)>4.7 THEN ! If >4.7 amps on center-fil, try to adjust
10322                          ! Re-187 magnet-value for drift.
10323    Center_peak(0,0) ! center Re-187, then restore original-peak mag-values
10324    L=L0
10326    OUTPUT 8 USING "2(4A,4Z),8A";"$0FJ",Magnet(L,2),"$0FK",Coarsemag(1)
10327    IF Subflag THEN PRINT USING "K,2/";"(Couldn't center the Re-187 peak)"
10328  END IF
10329  !
10330  ON ERROR GOTO 10356
10332  ASSIGN @Path1 TO "TYPE=INTERNAL"
10335  OUTPUT @Path1,Type;Runclass$,Niso,Mcoef(*),Magnet(*),Coarsemag(*),Peak_inter,Aside,Rf,Nuclide$(*),Normal(*),Inv,Interfere(*),
Hv
10338  Hv0=Hv
10341  OFF ERROR
10342  Whoop
10344  PRINT FNH$("*** ADJUSTED MAGNET-VALUES FOR "&TRIM$(Runclass$)&" STORED ***")
10347  WAIT 2
10350  SUBEXIT
```

```
10353  !
10356  PRINT USING "3/,10X,K";FNH$(" **** UNABLE TO STORE ON DISK ****")
10359  OFF ERROR
10362  Clunk
10365  WAIT 2
10368 SUBEND
10369 !
10370 !
10372 Data:SUB Data(Decay,Na$,Ln,Mbl,Nbl,Share_bkgrds,Dump_datagraf,Bad_pressure,INTEGER Niso,Block,Dy,Pb_4678,Run)! take a block of
 isotope-ratio data
10373  OPTION BASE 1
10376  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
10379  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
10382  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
10385  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
10388  COM /Magnet/ Mcoef(*),INTEGER Lll,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
10391  COM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel
10394  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
10397  COM /Data1/ Peak_in(*),Pk,Interfere(*),Normal(*),TOO,INTEGER Data_iso(*),Data_collector,Inv,Bmc_out,Next_run,Spike
10400  COM /Data2/ Acc(*),Lacc(*),Sigma$(*),Delta$(*),Aver(*),Last_aver(*),Last_ratio$(*),Ratio$(*)
10403  COM /Data3/ Bkrds(*),Bkrd_var(*),Resdecay(*),INTEGER Sample,N_prime,Nsets,Ref,Rf,Bkrd_rdgs(*),Bkrd_sigma(*)
10406  COM /Rat/ Slope,Int,Spikenumber,Sig,Peak_t(8,40),Timeconst_corr(8),Peak_height(8,40),Normrat_slope,Normrat_inter,Norm_inverte
d
10409  COM /Interfere/ Interfere0(4,3),Interf_nsecs,Interf_mon_cts(4,2),Interf_mon_time(4,2),INTEGER Long(4)
10412  INTEGER Bkrd_posn(12,2),Mag_pos(12),Mag_pos0(12),Ratio_isotope(12),Integr_time(8),Wait_time(8),K0,Pr,Specif_mon_pk(8),Dropout
s,N
10415  REAL Interf_corr_err(8),Peak_a(8),Peak(8),Sums(5)
10418  DIM Mul$(2)[4],U$[10],S$[20],S1$[7],Coef(4,1),Chi_square(40),Temp_aver(7)
10421!
10424  DATA 0,0,MILLIVOLTS,CUP,DALY,.2,0,0
10427!
10430 Chisq:DATA 0,0,5.99,7.81,9.49,11.1,12.6,14.1,15.5,16.9,18.3,19.7,21,22.4,23.7,25,26.3,27.6,28.7,30.1,31.4,32.7,33.9,35.2,36.4,
37.6,38.9,40.1,41.3,42.6
10433  DATA 43.8,44.9,46.2,47.4,48.6,49.8,51,52.2,53.4,54.6
10436   !
10439  GRAPHICS OFF
10442  OFF KNOB
10445  N=N_prime  ! don't allow value of N_prime as passed in Data3 to change
10448  OUTPUT 8 USING "4A,4Z";"$OFJ",FNIsomag(Mcoef(*),Data_iso(N)-.5) ! put magnet on a bkrd position
10451  FOR I=Auto*Full_auto*2 TO 19
10454    ON KEY I LABEL "" CALL Clunk
10457  NEXT I
10460  ON KEY 0 LABEL "   BMC" GOTO Bmc_out
10463  ON KEY 1 LABEL "   NEXT RUN" GOTO Next_run
10466  READ Pb4678_cycle,Daly_discr,U$,Mul$(*),Daly_discr,Bmc_out,Next_run,Chi_square(*)
10469  MAT Peak= Peak_in  ! Peak_in(*) in "DATA" subprogram is Peak(*) in Main program
10472  MAT Peak_in= (0)  ! only restore Peak_in array if complete data-block
10475  Interf_nsecs=Nsets  ! # seconds to spend on "short" interf-mon pks
10478  Spikenumber=Spike
10481  Number_peaks=N  ! # isotopes requested by user (doesn't incl. interf-mon pks)
10484  Number_of_sets=Nsets
10487  MAT Peak_a= Peak
10490  MAT Interf_mon_cts= (0)
10493  MAT Interfere0= Interfere ! determine which isotopes to monitor for isobaric interferences
10496  MAT Long= (0)
10499  MAT Interf_corr_err= (0)
10502  MAT Specif_mon_pk= (0)
10505  MAT Sigma$= ("")
10508  MAT Delta$= ("")
10511!
```

```
10514  FOR M=1 TO 4
10517    FOR I=1 TO N
10520      IF Interfere(M,2)=Data_iso(I) THEN 10535
10523    NEXT I
10526    Interfere0(M,1)=0 ! ignore isobaric interferences if none for the
10529    Interfere0(M,2)=0 ! specified isotopes.
10532    Interfere0(M,3)=0
10535  NEXT M
10538  FOR I=1 TO N
10541    Ratio_isotope(I)=Data_iso(I)  ! can include interf-mon isotopes
10544  NEXT I
10547  FOR M=1 TO 4
10550    FOR I=1 TO N
10553      IF Interfere(M,1)=Ratio_isotope(I) THEN
10556        Long(M)=I  ! order of monitor isotope in data-isotope list
10559        Interfere0(M,1)=0 ! so don't monitor before/after peaktops
10562        Specif_mon_pk(I)=1 ! if a specifically-requested interference-monitor peak
10565      END IF
10568    NEXT I
10571  NEXT M
10574  OUTPUT KBD;Clear$;
10577  ON KEY 9 LABEL "    ESCAPE" GOTO Exit_data
10580  IF NOT Auto THEN S1$="(of "&VAL$(Block+Nbl-Mbl+1)&")"
10583  IF Auto THEN S1$="(AUTO)"
10586  IF Nfils=2 THEN S$="SIDE-FIL.="&VAL$(Filament(2))&" AMPS"
10589  Tpress=TIMEDATE
10592  PRINTER IS CRT
10595  CALL Pressure(Mu,Mm$(*),1,Source_pressure,Tube_pressure)
10598  IMAGE  3/,80A,/,80A,/,21X,K,/,80A,/,80A,3/
10601  IF (Source_pressure)1.E-6) OR (Tube_pressure)1.E-7) THEN
10604    FOR Y=2 TO 1 STEP -1
10607      PRINTER IS Prtr(Y)
10610      IF Source_pressure)1.E-6 THEN
10613        PRINT USING 10598;RPT$("*",80),RPT$("*",80),FNBl$("SOURCE PRESSURE IS TOO HIGH ("&VAL$(Source_pressure)&")"),RPT$("*",8
0),RPT$("*",80)
10616      END IF
10619      IF Tube_pressure)1.E-7 THEN
10622        PRINT USING 10598;RPT$("*",80),RPT$("*",80),FNBl$("TUBE-PRESSURE IS TOO HIGH ("&VAL$(Tube_pressure)&")"),RPT$("*",80),R
PT$("*",80)
10625      END IF
10628    NEXT Y
10631    Superduperclunk
10634    IF Auto THEN
10637      IF (TIMEDATE-Tpress)/60)60 THEN  ! wait no more than 1 hour for pessure to improve (auto-running only)
10640        Bad_pressure=1
10643        GOTO Exit_data
10646      END IF
10649      PRINT "WAITING FOR ADEQUATE SOURCE AND TUBE PRESSURE..."
10652      Wait(TIMEDATE,60,Filament(Nfils),0,Auto,Full_auto)
10655      GOTO 10595
10658    ELSE
10661      GOTO Exit_data
10664    END IF
10667    OUTPUT KBD;Clear$;
10670  END IF
10673  Bad_pressure=0
10676  FOR Y=1 TO 2
10679    PRINTER IS Prtr(Y)
10682    IF Y=2 THEN PRINT USING "#,K";CHR$(27)&"&l8D" ! narrow line-spacing
10685    IF Y=2 THEN PRINT "SOURCE PRESSURE =";Source_pressure;TAB(48);"TUBE PRESSURE =";Tube_pressure
```

```
10688    IMAGE "BARREL#",3D," ",14X,"BLOCK# ",DD,2X,8A,4X,10A,8X,"RUN#",DD," ",/,"CENT. FIL.=",D.3D," AMPS",4X,20A,18X,DD," SETS
",/,"SAMPLE: ",50A,/
10691    PRINT USING 10688;Sample,1+Block,S1$,CHR$(128+3*Dy)&"**"&Mul$(1+Dy)&"**"&CHR$(128),Run,Filament(1),S$,Nsets,Na$
10694  NEXT Y
10697   ! if Comzer=1 then take backgrounds only for the least-intense peak
10700   ! - applies if cup data only, max-ratio<20, good pressure, not Pb4678.
10703  IF Peak(N)=0 THEN
10706    Superclunk
10709    PRINT USING "4/,K,2/,K,4/";"************** PEAK(N)=0 ERROR IN Data **************","################ PLEASE NOTIFY KEN #########
#####"
10712    Peak(N)=1
10715  END IF
10718  IF (Peak(1)/Peak(N)<20) AND (Dy=0) AND (Pb_4678=0) AND (Source_pressure<4.E-8) AND (Tube_pressure<6.E-9) THEN Comzer=1
10721  FOR I=1 TO N
10724    IF Ratio_isotope(I)=Ref THEN R=I  !R is order or ref-isot. in peak-switching
10727  NEXT I
10730 Setup_1: ! if Pb 204-206-207-208 block, take 6/4 data before & after 6/7/8 data
10733  IF Pb_4678 THEN
10736    Ratio_isotope(1)=206
10739    Ratio_isotope(2)=204
10742    Nsets=.7*Number_of_sets  ! but take only 70% of specified sets each time
10745    IF Nsets<5 THEN Nsets=5
10748    N=2   ! since only 206 & 204
10751    Number_peaks=2
10754    MAT Peak_a= (0)
10757    Peak_a(1)=Peak(R)  ! 206
10760    FOR I=1 TO 4
10763      IF Data_iso(I)=204 THEN Peak_a(2)=Peak(I)
10766    NEXT I
10769    Refpk=1
10772  ELSE
10775    Refpk=R
10778  END IF
10781 Setup_2:CALL Peaktime(Peak_a(*),Noise(*),Ions,Integr_time(*),Wait_time(*),N,Dy,(Refpk),Ratio_isotope(*))
10784  GCLEAR
10787  FOR J=1 TO 2
10790    PRINTER IS Prtr(J)
10793    IF J=2 THEN PRINT USING "#,K";CHR$(27)&"&l0D"
10796    PRINT USING "9A,12X,#";"ISOTOPES:"
10799    FOR I=1 TO N
10802      PRINT USING "3D,3X,#";Ratio_isotope(I)
10805      IF Specif_mon_pk(I)=1 THEN  ! if isotope is a specifically requested isobaric-interference monitor, use wait-t & integrat
ion-t of 3 seconds
10808        Wait_time(I)=2
10811        Integr_time(I)=3
10814      END IF
10817    NEXT I
10820    PRINT USING "/,18A,2X,#";"INTEGRATION-TIMES:"
10823    FOR I=1 TO N
10826      PRINT USING "3D,3X,#";Integr_time(I)
10829    NEXT I
10832    PRINT USING "/,11A,9X,#";"WAIT-TIMES:"
10835    FOR I=1 TO N
10838      PRINT USING "3D,3X,#";Wait_time(I)
10841    NEXT I
10844    PRINT
10847  NEXT J
10850  FOR R=1 TO N
10853    IF Ratio_isotope(R)=Ref THEN 10859 !R is order or ref-isot. in peak-switching
10856  NEXT R
```

```
10859  PRINTER IS Prtr(2);WIDTH (132)
10862  PRINT USING "#,K,/";CHR$(27)&"&k2S"  ! small print
10865  IF NOT Pb_4678 OR NOT Pb4678_cycle THEN
10868    PRINT "FOCUS:";TAB(19);  ! print focus-settings
10871    FOR I=1 TO 6+Nfils
10874      PRINT USING "3(K),3X,#";I,"-",Foc(I)
10877    NEXT I
10880    PRINT
10883  END IF
10886  Nn=N+4*(Pb_4678*Pb4678_cycle=0)  ! Expand to include possible interf-mon pks
10889  !
10892  FOR K=1 TO Nn  ! determine magnet-settings for peaktops & bkrds
10895    IF K>N THEN Ratio_isotope(K)=InterfereO(K-N,1)  ! add interference-monitor isotope to Ratio-Isotope list
10898    FOR J=1 TO Niso
10901      IF Ratio_isotope(K)=Magnet(J,1) THEN
10904        Mag_pos(K)=Magnet(J,2) ! peaktops
10907        !
10910        ! bkrds at 1/2 amu above & below peaktops
10913        Bkrd_posn(K,2)=Mag_pos(K)+(FNIsomag(Mcoef(*),Ratio_isotope(K)+.5)-FNIsomag(Mcoef(*),(Ratio_isotope(K)))) ! mag-setting f
or ABOVE bkrd
10916        Bkrd_posn(K,1)=Mag_pos(K)+(FNIsomag(Mcoef(*),Ratio_isotope(K)-.5)-FNIsomag(Mcoef(*),(Ratio_isotope(K)))) ! mag-setting
for BELOW bkrd
10919      END IF
10922    NEXT J
10925  NEXT K
10928  !
10931  Mu=Dy
10934  OUTPUT 8;Mm$(Mu,1)  ! 1-second integration
10937  DISP
10940  PRINT USING "K,#";"MAGNET:   ("&VAL$(Coarse)&")   "  ! print magnet-settings
10943  FOR I=1 TO Nn
10946    IF Ratio_isotope(I)<>0 THEN PRINT USING "3(K),3X,#";Ratio_isotope(I),"-",Mag_pos(I)
10949  NEXT I
10952  PRINT USING "/"
10955  K=0
10958  PRINTER IS Prtr(2);WIDTH (132)
10961  FOR I=1 TO Nn  ! Print headings for backgrounds
10964    FOR J=Number_peaks+1 TO I-1
10967      IF (Ratio_isotope(J)=Ratio_isotope(I)) OR (Ratio_isotope(I)=0) THEN 10979! avoid duplicate isotopes
10970    NEXT J
10973    K=K+1
10976    IF Ratio_isotope(I) THEN PRINT TAB(10*K-9);Ratio_isotope(I);
10979  NEXT I
10982  PRINT "      BACKGROUNDS: (cts per sec./std. dev.)"
10985 Bbkrd:!  take before-peaktop backgrounds
10988  IF NOT Share_bkgrds THEN !  If not sharing the last block's backgrounds
10991    CALL Backgrounds(Comzer,Number_peaks,Peak(*),R,Pb4678_cycle,Nn,0,N,Pb_4678,Integr_time(*),Bkrd_posn(*),Ratio_isotope(*),Coa
rsemag(1))
10994    IF Bmc_out OR Next_run OR Subflag>1 THEN Exit_data
10997  ELSE
11000    FOR I=1 TO Nn              ! Share last block's backgrounds
11003      FOR J=1 TO 2
11006        Bkrds(J,I)=Bkrds(J+2,I)
11009        Bkrd_rdgs(J,I)=Bkrd_rdgs(J+2,I)
11012      NEXT J
11015    NEXT I
11018  END IF
11021  OUTPUT 8;"$ICL"
11024  ENTER 8;In
11027  IF Pb_4678=0 OR Pb4678_cycle=0 THEN
```

```
11030    MAT Mag_pos0= Mag_pos
11033    CALL Interfere(0,In,Bkrds(*),Mag_pos0(*),(Number_peaks),Subflag,Bmc_out,Next_run)
11036    ON Subflag GOTO 11039,Bbkrd,Bbkrd,Exit_data
11039    IF Bmc_out OR Next_run THEN Exit_data
11042  END IF
11045  FOR M=1 TO 4 ! Put monitor isotope with data-isotope list if interference correction will be > .10% and monitor peak is >.5mV
/.05mV (Cup/Daly)
11048    P=Interf_mon_cts(M,1)/Mx(Mu) ! mV interference-monitor peak
11051    IF Interfere0(M,1) AND Long(M)=0 AND ((Mu=0)*P)>.5) OR (Mu*P)>.05) THEN
11054      FOR I=1 TO N
11057        IF Interfere0(M,2)=Ratio_isotope(I) AND N<8 THEN
11060          Checked=0
11063          FOR J=1 TO M-1! only monitor an interference peak once per set
11066            IF (Interfere(J,1)=Interfere(M,1)) AND Long(J) THEN Checked=1
11069          NEXT J
11072          Percent_corr=100*(Interf_mon_cts(M,1)/Mx(Mu))*Interfere0(M,3)/Peak(I)! % correction from isobaric interference
11075          IF Percent_corr>.10 AND NOT Checked THEN ! if correction >.10% then use "long" interference monitor
11078!
11081            N=1+N! increment the number of isotopes in the set
11084            Long(M)=N
11087            FOR P=1 TO 2! allocate interf-mon bkrds to correct main-pk order
11090              Bkrds(P,N)=Bkrds(P,Number_peaks+M)
11093              Bkrd_rdgs(P,N)=Bkrd_rdgs(P,Number_peaks+M)
11096            NEXT P
11099            Peak(N)=Peak(N-1)
11102            Ratio_isotope(N)=Interfere(M,1)
11105            Wait_time(N)=2
11108            Integr_time(N)=3+(Percent_corr>.2)+2*(Percent_corr>.4)! integrate for 3 seconds if correction<.2%, 4 if between .2%
 and .4%, 6 if >.4%
11111            Interfere0(M,1)=0
11114            Interfere0(M,2)=0
11117            FOR K=1 TO Niso
11120              IF Magnet(K,1)=Ratio_isotope(N) THEN Mag_pos(N)=Magnet(k,2)
11123            NEXT K
11126          END IF
11129        END IF
11132      NEXT I
11135    END IF
11138  NEXT M
11141  Short_interf=0
11144  FOR M=1 TO 4  ! count # of "short"-monitored interf-monitor pks
11147    IF Interfere0(M,1) THEN Short_interf=1+Short_interf
11150  NEXT M
11153   !
11156 Pks:PRINTER IS CRT
11159  OUTPUT KBD;Clear$;
11162  IF NOT Pb_4678 OR Pb4678_cycle=1 THEN
11165    Pmax=1.2*Peak(1)
11168  ELSE
11171    Pmax=1.2*Peak_a(1)
11174  END IF
11177  Axes(0,100,36,86,0,(20*Short_interf+Nsets*(SUM(Integr_time)+SUM(Wait_time)))/60,0,Pmax,"MINUTES","mV BEAM",0,-1,0)
11180  PRINT USING 10688;Sample,1+Block,S1$,"**"&Mul$(1+Mu)&"**",Run,Filament(1),S$,Nsets,Na$
11183  CALL Peaks(R,Peak_t(*),K,In,Peak_height(*),Pmax,Temp_aver(*),N,Coarsemag(1),Mag_pos(*),Integr_time(*),Wait_time(*),Ratio_isot
ope(*),Dropouts)
11186  IF Bmc_out OR Next_run THEN Exit_data
11189  ON Subflag GOTO Abkrd,Over_10volts,Exit_data,Exit_data
11192!
11195 Abkrd:! take after-peaktop backgrounds
11198  PRINTER IS Prtr(2);WIDTH (132)
```

```
11201  CALL Backgrounds(Comzer,Number_peaks,Peak(*),R,Pb4678_cycle,Nn,2,N,Pb_4678,Integr_time(*),Bkrd_posn(*),Ratio_isotope(*),Coars
emag(1))
11204  PRINT USING "#,K";CHR$(27)&"&k0S"  ! normal print
11207  IF Bmc_out OR Next_run OR Subflag>1 THEN Exit_data
11210  !
11213  FOR M=1 TO 4 ! Shift interf-mon bkrds for spec.-mon pks to correct order
11216    IF Long(M) THEN
11219      IF NOT Specif_mon_pk(Long(M)) THEN ! if a not specifically-requested interference-monitor peak
11222        FOR P=3 TO 4
11225          Bkrds(P,Long(M))=Bkrds(P,Number_peaks+M)
11228          Bkrd_rdgs(P,Long(M))=Bkrd_rdgs(P,Number_peaks+M)
11231        NEXT P
11234      END IF
11237    END IF
11240  NEXT M
11243  PRINTER IS CRT
11246  !
11249  MAT Interf_corr_err= (0)
11252  FOR M=1 TO 4
11255    FOR J=1 TO 4
11258      IF (Interfere(J,1)=Interfere(M,1)) AND Long(J) THEN Long(M)=Long(J)
11261    NEXT J
11264  NEXT M
11267  IF NOT Pb_4678*Pb4678_cycle THEN
11270    CALL Interfere(2,In,Bkrds(*),Mag_pos0(*),(Number_peaks),Subflag,Bmc_out,Next_run)
11273    IF Subflag=2 OR Subflag=3 THEN Bbkrd
11276    IF Bmc_out OR Next_run OR Subflag=4 THEN Exit_data
11279  END IF
11282  DISP
11285  Sum_bkrd_rdgs=0
11288  PRINTER IS Prtr(2)
11291  FOR I=1 TO N  ! calculate average backgrounds
11294    Bkgrd=0
11297    Bkrd_var(I)=0
11300    FOR K=1 TO 4
11303      Sum_bkrd_rdgs=Sum_bkrd_rdgs+Bkrd_rdgs(K,I)  ! total# bkrd rdgs used for Ith peak
11306      Bkgrd=Bkgrd+Bkrds(K,I)/4  ! average background in counts/second
11309      Bkrd_var(I)=Bkrd_var(I)+Noise(Mu)^2/(4*Bkrd_rdgs(K,I))  ! Average background variance of the mean, in  (mV/sec)^2, for It
h peak
11312    NEXT K
11315    FOR J=1 TO Nsets
11318      Peak_height(I,J)=Peak_height(I,J)-Bkgrd  ! subtract backgrounds
11321    NEXT J
11324  NEXT I
11327  MAT Timeconst_corr= (0)
11330  IF Mu=0 THEN CALL Timeconst_corr(Timeconst_corr(*),Resdecay(*),Peak_height(*),N,Integr_time(*),Wait_time(*))
11333  FOR M=1 TO 4  ! correct for isobaric interferences
11336    IF Long(M)=0 THEN
11339      IF Interfere0(M,1)=0 THEN 11453  ! " short" isobaric interf. monitor (linear interp)
11342      Coef(2,1)=(Interf_mon_cts(M,2)-Interf_mon_cts(M,1))/(Interf_mon_time(M,2)-Interf_mon_time(M,1)) ! slope of interf-monitor
 peak decay
11345      Coef(1,1)=Interf_mon_cts(M,1)-Coef(2,1)*Interf_mon_time(M,1)  ! intercept of interf-monitor peak decay
11348      Coef(3,1)=0
11351      Coef(4,1)=0
11354    ELSE
11357      CALL Cubic(Coef(*),Peak_t(*),Peak_height(*),Nsets,Long(M))! "Long" isobaric-interference monitor (cubic interp)
11360    END IF
11363    FOR I=1 TO N
11366      IF Interfere(M,2)=Ratio_isotope(I) THEN
11369        Av_monitor_pk=0
```

```
11372       Av_interf_corr=0
11375       Npcor=0
11378       FOR J=1 TO Nsets
11381         T=Peak_t(I,J)
11384         Mon_pk=Coef(1,1)+T*(Coef(2,1)+T*(Coef(3,1)+T*Coef(4,1)))! predicted monitor-pk intensity
11387         Interf_pk=Mon_pk*Interfere(M,3)! predicted interfering-pk intensity
11390         Av_monitor_pk=Av_monitor_pk+Mon_pk/Nsets! average monitor-peak, in counts/second
11393         Peak_height(I,J)=Peak_height(I,J)-Interf_pk! correct for isobaric interference
11396         Pcor=100*Interf_pk/Peak_height(I,J)! average % correction to Peak_height(I,J)
11399         !Test for very large correction due to being knocked off peak by seeing if correction for last peak was within a fact
or of 40 (>2mV only)
11402         IF J>1 AND (Pcor/(Lpcor+(Lpcor=0))<40 OR Peak_height(I,J)/Mx(Mu)).02*Mu+2*(Mu=0)) THEN
11405           Av_interf_corr=Av_interf_corr+Pcor
11408           Npcor=1+Npcor
11411           Lpcor=Pcor
11414         END IF
11417       NEXT J
11420       Av_interf_corr=Av_interf_corr/(Npcor+(Npcor=0))
11423       IF Av_monitor_pk THEN
11426         PRINT "Average";Interfere(M,1);TAB(13);"peak was ";DROUND(Av_monitor_pk/Mx(Mu),3);"mV";
11429         PRINT TAB(35);"Correction on";Ratio_isotope(I);TAB(54);"= ";DROUND(Av_interf_corr,3);"%"
11432         D=SQR(2)*Av_interf_corr*Noise(Mu)*Mx(Mu)/Av_monitor_pk
11435         IF Long(M)=0 THEN D=D/SQR(Interf_nsecs) ! %error from isobaric-interference correction
11438         IF Long(M) THEN D=D/SQR(Integr_time(Long(M))*Nsets) ! ditto
11441         Interf_corr_err(I)=SQR(Interf_corr_err(I)^2+D^2+(.02*Av_interf_corr)^2) ! % error from isobar-interf correction, incl
 2% +/- in interf ratio
11444       END IF
11447     END IF
11450   NEXT I
11453 NEXT M
11456 MAT Sums= (0)
11459 FOR J=1 TO Nsets  ! for calculating average beam-decay/growth
11462   CALL Sums(Sums(*),1,(Peak_height(R,J)),Peak_t(R,J),0)
11465 NEXT J
11468 CALL Slope(Sums(*),Decay,Int,S0,S9,Nsets)
11471 Pk=Sums(1)/Nsets/Mx(Mu)
11474 Decay=Decay*Nsets*6.E+5/Sums(1)
11477 Whoop
11480 PRINT DROUND(Pk,3);"mV";Ref;
11483 FOR I=1 TO N
11486   IF I<>R THEN
11489     Ix=I-(I>R)
11492     PRINT TAB(15*Ix);DROUND(Pk/Temp_aver(Ix)^Inv,3);"mV";Ratio_isotope(I);
11495   END IF
11498 NEXT I
11501 PRINT
11504 IF Dropouts THEN PRINT USING "3(K)";"<<<<<<< ",Dropouts," GPIO dropout(s) in this block >>>>>>>"
11507 PRINT RPT$("-",80)
11510   !
11513 Doratios:I=Pb_4678*(Pb4678_cycle=1)   ! calculate isotope ratios
11516 ! Pb4678_cycle (for a Pb-204-206-207-208 block) is 1 after 1st 6/4 done, 2 after 6/7/8 done, 3 after last 6/4 done
11519 Pb4678_cycle=Pb4678_cycle+1
11522 Ratio_count=0
11525 Normrat_acc=0
11528 Printed=0
11531   !
11534 Call_ratiocalc:FOR I0=1 TO N
11537   IF I0=R THEN Calc_nextratio
11540   I=I+1
11543! don't calculate a ratio if peak I0 is the ref-pk, or if an unspiked, fractionation-normalizable run until the normalizing rat
```

```
io has been calculated.
11546    IF (NOT Spike)*Normal(1)*(Ratio_count*(Ratio_isotope(IO)=Normal(1)) OR (NOT Ratio_count)*(Ratio_isotope(IO)<>Normal(1))) TH
EN Calc_nextratio
11549    PRINTER IS Prtr(2)
11552    CALL Ratiocalc(I,IO,Calc_inverted,R,Mass_discr,Ratio_count,Pb4678_cycle,Normal(*),N_sets_used,Decay,Nsets,Pb_4678,Ref,Inv,M
u,Ratio_isotope(*))
11555     !
11558 Final:Av=ABS(Aver(I))
11561    Temp=(N_sets_used-1)/N_sets_used*Noise(Mu)^2*(Av^(2*Inv)/Integr_time(IO)+1/Integr_time(R))/Pk^2+(Av^Inv/Integr_time(IO)+1/I
ntegr_time(R))/(Pk*Ions)
11564    Tsig=100*SQR(Temp)
11567    IF Comzer=0 THEN Bkrd_uncert=100*SQR(Bkrd_var(IO)*Av^(2*Inv)+Bkrd_var(R))/Pk
11570    IF Comzer=1 THEN Bkrd_uncert=100*SQR(Bkrd_var(N)*((Av^Inv-1)/Pk)^2)
11573 ! Bkrd_uncert = %error in ratio due to background uncertainty
11576    Ierr=SQR(Interf_corr_err(R)^2+Interf_corr_err(IO)^2)   ! % error due to isobaric-interference corrections
11579    Acc(I)=SQR(MAX((Sig),(Tsig))^2/N_sets_used+Bkrd_uncert^2+Ierr^2+(Normrat_acc*(Ref-Ratio_isotope(IO))/(Ref-Normal(1)))^2)  !
 sigmamean(%)
11582    IF Pb_4678 THEN
11585      ON Pb4678_cycle GOTO 11708,11615,11591
11588    ! Combine values for 1st & last 206/204
11591      N_sets_used=N_sets_used+(First64_nsets+2)/2
11594      Recip_var=1/Acc(1)^2+1/First64_sigmean^2
11597      Aver(1)=(Av/Acc(1)^2+First_64/First64_sigmean^2)/Recip_var
11600      Acc(1)=1/SQR(Recip_var)
11603      Sig=1/SQR((1/Sig^2+1/First64_sig^2)/2)
11606      Tsig=SQR((Tsig^2+First64_tsig^2)/2)
11609    END IF
11612     !
11615    FOR K=1 TO Ln-1  ! Calculate percent change in ratios, put in parentheses if within theoretical
11618      IF Block AND (Last_ratio$(K)=Ratio$(I)) AND Last_aver(k) THEN CALL Delta(Delta$(I),Aver(I),Last_aver(K),Acc(I),Lacc(K))
11621    NEXT K
11624    Sigma$(I)[1,8]=TRIM$(VAL$(DROUND(Sig,2)))
11627    Sigma$(I)=TRIM$(Sigma$(I))
11630    Within_theor=(Sig/Tsig<SQR(Chi_square(INT(N_sets_used))/(INT(N_sets_used)-1)))
11633    IF Within_theor THEN Sigma$(I)="("&Sigma$(I)&")"
11636    GRAPHICS OFF
11639    IF NOT Printed THEN OUTPUT KBD;Clear$;
11642    Printed=1
11645    FOR Y=1 TO 2-Pb_4678*(Pb4678_cycle=1)
11648      PRINTER IS Prtr(Y);WIDTH (132)
11651      T=5*(Y=2)
11654      PRINT "  AVERAGE";TAB(16);"AVERAGE";TAB(29);"SIGMA%";TAB(40);"SIGMA%";TAB(50);"SIGMA%";TAB(60);"SIGMA"
11657      IF Y=2 THEN PRINT USING "#,K";CHR$(27)&"&l6D"  ! normal line-spacing
11660      PRINT "  "&Ratio$(I);TAB(16+T);FNInvert$(Ratio$(I));TAB(30+T);"OBS.";TAB(40+T);"THEOR.";TAB(50+T);"MEAN";TAB(60+T);"MEAN"
;TAB(70+T);"DELTA%"
11663      IF Y=1 AND N<4 THEN PRINT
11666      PRINT TAB(2);DROUND(Aver(I),6);TAB(15);DROUND(1/Aver(I),6);TAB(29);Sigma$(I);TAB(39);DROUND(Tsig,2);
11669      PRINT TAB(49);DROUND(Acc(I),2);TAB(59);DROUND(Acc(I)*Aver(I)/100,2);TAB(70);Delta$(I)
11672      IF Y=2 AND I=N-1 THEN PRINT USING "#,K";CHR$(27)&"&l8D"
11675      PRINT RPT$("-",80)
11678    NEXT Y
11681    Ratio_count=Ratio_count+1
11684    IF (Ratio_count=1) AND Normal(1) AND Spike=0 THEN
11687      Normrat_slope=Slope  ! Linear-regression slope of normalizing ratio
11690      Normrat_inter=Int    !  "       '      intercept    "       "
11693      Norm_inverted=Calc_inverted
11696      Normrat_acc=Acc(I)  ! uncertainty of  normalizing ratio
11699      I=0
11702      GOTO Call_ratiocalc
11705    END IF
```

```
11708    PRINTER IS CRT
11711 Calc_nextratio:NEXT IO
11714  !
11717  IF Pb_4678 AND Pb4678_cycle<3 THEN
11720    Center_peak(1,Rf)! re-center the 206 peak & recalibrate mag-settings
11723    IF Subflag=1 THEN Bmc_out
11726    SELECT Pb4678_cycle
11729    CASE 1  ! after 1st 206/204 sets
11732      PRINT USING 10688;Sample,1+Block,Si$,"**"&Mul$(1+Mu)&"**",Run,Filament(1),S$,Nsets,Na$
11735      MAT Peak_a= (0)
11738      J=0
11741      FOR I=1 TO 4
11744        IF Data_iso(I)<>204 THEN
11747          J=J+1
11750          Ratio_isotope(J)=Data_iso(I)
11753          Peak_a(J)=Peak(I)
11756        END IF
11759      NEXT I
11762      N=3
11765      Number_peaks=3
11768      Nsets=Number_of_sets
11771      First_64=Aver(1)
11774      First64_sigmean=Acc(1)
11777      First64_sig=Sig
11780      First64_tsig=Tsig
11783      First64_nsets=N_sets_used
11786      FOR I=1 TO 3! find order of 206 & 207 peaks
11789        IF Ratio_isotope(I)=206 THEN Refpk=I
11792!     IF Ratio_isotope(I)=207 THEN Peak_207=I
11795      NEXT I
11798      R=Refpk
11801      FOR K=1 TO 2 ! share 206 bkgrds from 6-4 to 6-7-8 if 6/7 ratio not too large
11804        Bkrds(K,Refpk)=Bkrds(2+K,1)
11807        Bkrd_rdgs(K,Refpk)=Bkrd_rdgs(2+K,1)
11810        Bkrd_sigma(K,Refpk)=Bkrd_sigma(2+K,1)
11813      NEXT K
11816      PRINTER IS Prtr(2)
11819      PRINT RPT$("-",80)
11822      PRINTER IS CRT
11825      GOTO Setup_2
11828!
11831    CASE 2! after 206-207-208 sets
11834      FOR K=1 TO 2
11837        Bkrds(K,1)=Bkrds(2+K,Refpk)
11840        Bkrd_rdgs(K,1)=Bkrd_rdgs(2+K,Refpk)
11843        Bkrd_sigma(K,1)=Bkrd_sigma(2+K,Refpk)
11846      NEXT K
11849      IF Dump_datagraf THEN DUMP GRAPHICS ! only dump 206-207-208 graphics
11852      GOTO Setup_1
11855    END SELECT
11858  END IF
11861!
11864  IF Dump_datagraf AND NOT Pb_4678 THEN DUMP GRAPHICS
11867  MAT Last_aver= Aver
11870  MAT Lacc= Acc
11873  MAT Last_ratio$= Ratio$
11876  L=Rf
11879  Data_collector=Mu
11882  Block=1+Block
11885 Exit_data:Ln=N_prime*Pb_4678+N*(Pb_4678=0) ! include any added interf.-mon
```

```
11888  Nsets=Number_of_sets                    ! pks if not a Pb-4678 block
11891  MAT Peak_in= Peak ! Restore Peak_in array
11894  PRINTER IS CRT
11897  SUBEXIT
11900   !
11903 Over_10volts:IF K<4 THEN Exit_data ! Stop peak-switching if beam > 10V
11906  Nsets=K-1
11909  GOTO Abkrd
11912  !
11915 Next_run:Next_run=1
11918  GOTO Exit_data
11921  !
11924 Bmc_out:Bmc_out=1
11927  IF Auto AND SUM(Peak_in)=0 AND SUM(Peak)>0 THEN MAT Peak_in= Peak
11930  ! to solve early-exit zeroing of Peak array in auto-running
11933  GOTO Exit_data
11936 SUBEND ! ----------------------------------------------------------------
11939 !
11942 !
11945 Sums:SUB Sums(Sums(*),G,Y,X,INTEGER N)
11948! Add (G=1) or subtract (G=-1) sums, sums-of-squares, & cross-product sums
11951  IF (N<3) AND (G<0) THEN SUBEXIT
11954  Sums(1)=Sums(1)+G*Y    ! sum of Y
11957  Sums(2)=Sums(2)+G*Y*Y  ! sum of squares of Y
11960  Sums(3)=Sums(3)+G*X    ! sum of X
11963  Sums(4)=Sums(4)+G*X*X  ! sum of squares of X
11966  Sums(5)=Sums(5)+G*X*Y  ! cross-product sums
11969  N=N-(G<0)
11972 SUBEND ! ----------------------------------------------------------------
11975!
11978!
11981 Slope:SUB Slope(Sums(*),Slope,Intercept,Std_dev_pts,Std_dev_slope,INTEGER N) ! calculate linear-regression
11984  Slope=(N*Sums(5)-Sums(3)*Sums(1))/(N*Sums(4)-Sums(3)^2)
11987  Intercept=(Sums(1)-Slope*Sums(3))/N
11990  S=(Sums(2)-Intercept*Sums(1)-Slope*Sums(5))/(N-2)
11993  Std_dev_pts=SQR(S*(S>0))              ! std deviation of pts about line
11996  S=Sums(4)-Sums(3)^2/N
11999  Std_dev_slope=Std_dev_pts/SQR(S*(S>0))  ! std deviation of slope
12002 SUBEND ! ----------------------------------------------------------------
12005!
12008!
12011 Runvariables:SUB Runvariables(Runvar0(*),Run_name$(*),Runtype$(*),Sample_name$(*),INTEGER Run_order(*),Run_iso(*),Auto)
12014! Change/edit/add/replace run-variables or std-run variables
12017  OPTION BASE 1
12020  COM /Keyboard/ Cn$,C1$,Cb$,Cu$,Q$,Clear$
12023  COM /Auto_form1/ Lastresponse$(-4:27)[50],Nresp(-4:27),Last_runtype$[8],Normal(2),Response$(-4:27)[50],Prompt$(-4:27)[36]
12026  COM /Auto_form2/ Spike,INTEGER Isotope(8),Niso,Rf,Magnet(0:24,2)
12029  DIM Stdvar(32,27),Std_name$(32)[10],Stdtype$(32)[8],A$[20],Std$[24],Run$[18],Element$(20)[10],S$(9)[160],S_name$(32)[50],Ogas
(32)
12032  INTEGER Std_iso(32,8)
12035  OUTPUT KBD;CHR$(255)&CHR$(75);
12038  Got_runvars=0
12041  Got_stdvars=0
12044  !
12047 Again:OFF KEY
12050  OFF KBD
12053  OFF KNOB
12056  A=KNOBX
12059  GRAPHICS OFF
12062  Last_runtype$=""
```

```
12065  Prompt$(1)="DEFINE A NEW SET OF "&FNUn$("RUN")&" VARIABLES"
12068  Prompt$(2)="EDIT or VIEW "&FNUn$("RUN")&" VARIABLES"
12071  Prompt$(5)="WHAT "&FNUn$("ARE")&" RUN & STD-RUN VARIABLES?"
12074  Prompt$(6)="HOW TO DEFINE OUTGASSING VARIABLES?"
12077  Prompt$(7)="EDIT or VIEW "&FNUn$("STD-RUN")&" VARIABLES"
12080  Prompt$(3)="ADD TO "&FNUn$("RUN")&" VARIABLES"
12083  Prompt$(8)="ADD TO "&FNUn$("STD-RUN")&" VARIABLES"
12086  Prompt$(4)="PRINTOUT ALL "&FNUn$("RUN")&" VARIABLES"
12089  Prompt$(9)="PRINTOUT ALL "&FNUn$("STD-RUN")&" VARIABLES"
12092  IF NOT Auto THEN Prompt$(10)="RETURN TO BMC"
12095  IF Auto THEN Prompt$(10)="BEGIN AUTOMATIC RUNNING"
12098!
12101  ON KEY 0 LABEL " NEW RUN VARS" GOTO New_runvars
12104  ON KEY 1 LABEL "EDIT RUN VARS" GOTO Edit_runvars
12107  ON KEY 4 LABEL "    HELP!" GOTO Help
12110  ON KEY 5 LABEL "   OUTGAS?" GOTO Outgas
12113  ON KEY 6 LABEL " "&Q$&"  "&"STDRUN VARS" GOTO Edit_stdvars
12116  ON KEY 2 LABEL "ADD TO RUNVARS" GOTO Add_runvars
12119  ON KEY 7 LABEL " "&Q$&"  "&Q$&"  "&"STDRUN VARS" GOTO Add_stdvars
12122  ON KEY 3 LABEL "PRINT RUNVARS" GOTO Print_runvars
12125  ON KEY 8 LABEL "  "&Q$&"   "&"STDRUN VARS" GOTO Print_stdvars
12128  IF NOT Auto THEN ON KEY 9 LABEL "    ESCAPE" GOTO 12143
12131  IF Auto THEN ON KEY 9 LABEL "  BEGIN RUNS" GOTO 12143
12134  CALL Keymenu(Prompt$(*))
12137  GOTO 12137
12140 !
12143  IF Runchange THEN GOSUB Store_runvars
12146  IF Stdchange THEN GOSUB Store_stdvars
12149  SUBEXIT
12152 !
12155 Help:RESTORE 12158
12158  DATA RUN-variables define how the samples will be run during an automatic run-sequence.  You must assign each sample its ow
n RUN variables.$
12161  DATA "STD-RUN variables are examples of typical RUN-variables that you can easily assign as the RUN variables.  This wa
y, instead"
12164  DATA  "of having to fill in all of the entries for the RUN variables for a sample, you can just specify"
12167  DATA "the STD-RUN variables that best match what you want, and then edit just a few of the values.$"
12170  DATA "To use a particular set of STD-RUN variables as RUN-variables, enter the STD-RUN number as the RUN-ELEMENT, and
 enclose"
12173  DATA "it in asterisks (you may need a printout of the STD-run variables before you do this)."
12176  DATA "For example, *12*, to use values from STD-RUN number 12.$"
12179  DATA "To get a display showing the names & isotopes of the STD-RUN variables, enter  *?*  as the RUN-ELEMENT."
12182  REDIM S$(8)
12185  READ S$(*)
12188  Helpscreen(S$(*))
12191  DATA "$$$$$$$You can get specific HELP for any parameter in the RUN VARIABLES form by pressing CTRL-H instead of entering the
 parameter."
12194  REDIM S$(1)
12197  READ S$(*)
12200  Helpscreen(S$(*))
12203  GOTO Again
12206  !
12209 Outgas:RESTORE 12212
12212  DATA "To define RUN-variables for automatic outgassing (no data-taking), enter the"
12215  DATA "run-type as either OUTGAS or P-OUTGAS.  To outgas the sample in front of the"
12218  DATA "source (normal running position), use OUTGAS.$"
12221  DATA "To outgas a sample in the preheat position, use P-OUTGAS.$"
12224  DATA "For preheat-position outgassing, enter the barrel-number of the actual sample to be outgassed."
12227  REDIM S$(5)
12230  READ S$(*)
```

```
12233  Helpscreen(S$(*))
12236  GOTO Again
12239  !
12242 Print_stdvars:Std=1
12245  IF NOT Got_stdvars THEN GOSUB Get_stdvars
12248  GOSUB Printvars
12251  GOTO Again
12254 Print_runvars:Std=0
12257  IF NOT Got_runvars THEN GOSUB Get_runvars
12260  GOSUB Printvars
12263  GOTO Again
12266!
12269 New_runvars:Run=0  ! define a completely new set of run-variables
12272  Std=0
12275  MAT Runvar0= (0)
12278  MAT Run_order= (0)
12281  MAT Lastresponse$= ("")
12284  Last_runtype$="??"
12287  WHILE Esc=0 AND Run<=32
12290    Run=1+Run
12293    Lastresponse$(0)="??"
12296    Lastresponse$(-3)="??"
12299    Auto_form(Run,0,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
12302    IF Esc THEN
12305      Esc=0
12308      Run=Run-1
12311      GOTO 12323
12314    END IF
12317    GOSUB Extract_runvars
12320  END WHILE
12323  IF Run=0 THEN Again
12326  Got_runvars=1
12329  GOSUB Printvars
12332  GOTO Again
12335!
12338 Edit_runvars:! edit or view the run-variables for a single run
12341  OUTPUT KBD;Clear$;
12344  OFF KEY
12347  Run=0
12350  Std=0
12353  IF NOT Got_runvars THEN GOSUB Get_runvars
12356  INPUT "WHICH RUN-NUMBER DO YOU WISH TO EDIT or VIEW?    (press CONTINUE to escape)",Run
12359  IF Run<1 OR Run>32 THEN Again
12362  IF Run_order(1)=0 THEN GOSUB Get_runvars
12365  GOSUB Convert_runvars
12368  Auto_form(Run,0,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
12371  IF Esc THEN
12374    Esc=0
12377    GOTO Edit_runvars
12380  END IF
12383  GOSUB Extract_runvars
12386  GOTO Edit_runvars
12389!
12392 Edit_stdvars:! edit/view the standard-run variables
12395  Run=0
12398  Std=1
12401  OUTPUT KBD;Clear$;
12404  OFF KEY
12407  INPUT "WHICH STANDARD-RUN NUMBER TO YOU WISH TO EDIT or VIEW (press CONTINUE to escape)",Run
12410  IF Run<1 OR Run>32 THEN Again
```

```
12413  IF NOT Got_stdvars THEN GOSUB Get_stdvars
12416  FOR I=1 TO 32
12419    IF Stdvar(I,1)=0 THEN 12425
12422  NEXT I
12425  Stdmax=I-1
12428  IF Run>Stdmax THEN
12431    Clunk
12434    DISP Ci$&" SORRY, ONLY";Stdmax;"STANDARD RUNS HAVE BEEN DEFINED "&Cn$
12437    WAIT 2
12440    GOTO Again
12443  END IF
12446  GOSUB Convert_stdvars
12449  Auto_form(Run,1,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
12452  IF Esc THEN
12455    Esc=0
12458    GOTO Edit_stdvars
12461  END IF
12464  GOSUB Extract_stdvars
12467  GOTO Edit_stdvars
12470!
12473 Add_runvars:Std=0 ! add to a set of existing run-variables
12476  OFF KEY
12479  IF NOT Got_runvars THEN GOSUB Get_runvars
12482  Lastresponse$(0)="??"
12485  Lastresponse$(-3)="??"
12488  Run=1
12491  WHILE Run_order(Run) AND Run<=32
12494    Run=1+Run
12497  END WHILE
12500  IF Run>32 THEN
12503    DISP FNH$("ALL 32 POSSIBLE RUNS ARE DEFINED ON THIS DATA-DISK")
12506    Clunk
12509    WAIT 2
12512    GOTO Again
12515  END IF
12518!
12521  Auto_form(Run,0,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
12524  IF Esc THEN
12527    Esc=0
12530    GOTO Again
12533  END IF
12536  GOSUB Extract_runvars
12539  GOTO 12482
12542!
12545 Add_stdvars:Std=1 ! add to existing standard-run variables
12548  OFF KEY
12551  IF NOT Got_stdvars THEN GOSUB Get_stdvars
12554  Run=1
12557  WHILE Stdvar(Run,1) AND Run<=32
12560    Run=1+Run
12563  END WHILE
12566  IF Run>32 THEN
12569    DISP FNH$(" ALL 32 POSSIBLE STANDARD-RUNS ARE DEFINED ")
12572    Clunk
12575    WAIT 2
12578    GOTO Again
12581  END IF
12584!
12587  MAT Lastresponse$= ("")
12590  Auto_form(Run,1,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
```

```
12593  IF Esc THEN
12596    Esc=0
12599    GOTO Again
12602  END IF
12605  GOSUB Extract_stdvars
12608  GOTO Again
12611!
12614 Printvars:! printout all run or std-run variables
12617  PRINTER IS 701;WIDTH (132)
12620  OFF KEY
12623  ON KEY 9 LABEL "    ESCAPE" GOTO 12887
12626  PRINT USING "2(K)";CHR$(27)&"&k2S",CHR$(27)&"&l6D" ! small print @ 6 lines/inch
12629  IF NOT Std THEN  ! runvar printout
12632    PRINT USING "2/,K,2/,7A,4X,11A,30X,8A,30X,8A,/";"RUN-VARIABLES: "&RPT$("*",104),"BARREL#","SAMPLE NAME","ISOTOPES","ELEMENT
"
12635    FOR Run=1 TO 32
12638      I=Run_order(Run) ! Barrel# for this run
12641      Ogas(Run)=(POS(UPC$(Runtype$(Run)[1,8]),"OUTGAS")<>0)
12644      IF I=0 THEN 12683
12647      PRINT TAB(2);I;TAB(8);
12650      IF Run_name$(Run)<>"" THEN PRINT Run_name$(Run);
12653      IF Run_name$(Run)="" THEN PRINT Sample_name$(I);
12656      IF Ogas(Run) THEN
12659        PRINT TAB(54);"****";
12662      ELSE
12665        FOR J=1 TO 8! printout run isotopes
12668          IF Run_iso(Run,J) THEN PRINT TAB(46+5*J);Run_iso(Run,J);
12671        NEXT J
12674      END IF
12677      PRINT TAB(92);Runtype$(Run)
12680    NEXT Run
12683    PRINT USING "3/,5X,8A,17X,13A";"VARIABLE","BARREL-NUMBER"
12686  ELSE ! standard-runvar printout
12689    PRINT USING "2/,K,2/,13A,4X,4A,8X,4A,14X,8A,/";"STANDARD-RUN VARIABLES: "&RPT$("*",100),"STANDARD-RUN#","ELEMENT","NAME","I
SOTOPES"
12692    FOR Srun=1 TO 32
12695      IF Stdvar(Srun,1) THEN
12698        PRINT TAB(6);Srun;TAB(18);UPC$(Stdtype$(Srun));TAB(29);Std_name$(Srun);
12701        IF POS(UPC$(Stdtype$(Srun)[1,8]),"OUTGAS") THEN
12704          PRINT TAB(50);"****";
12707        ELSE
12710          FOR J=1 TO 8
12713            IF Std_iso(Srun,J) THEN PRINT TAB(40+6*J);Std_iso(Srun,J);
12716          NEXT J
12719        END IF
12722      END IF
12725    NEXT Srun
12728    PRINT USING "3/,2X,8A,21X,K";"VARIABLE","STANDARD-RUN NUMBER"
12731  END IF
12734 !
12737  FOR Group=0 TO 1
12740    IF NOT Std AND Group AND NOT Run_order(17) THEN 12887
12743    IF Std AND Group AND NOT Stdvar(17,1) THEN 12887
12746    FOR Run=1+16*Group TO 16+16*Group
12749      R=Run-16*Group
12752      IF NOT Std AND Run_order(Run) THEN PRINT TAB(24+6*R);Run_order(Run);
12755      IF Std AND Stdvar(Run,1) THEN PRINT TAB(24+6*R);Run;
12758    NEXT Run
12761    PRINT USING "/"
12764!
```

```
12767   DATA SINGLE(1)/TRIPLE(3),FOCUSING ISOTOPE,CENTER-FIL BEAM (U),INITIAL CF-CURR.(A),DALY (0/1/2),CURRENT-1,RATE-1,WAIT-1 (MIN
.),CURRENT-2,RATE-2
12770   DATA WAIT-2 (MIN.),DATA WAIT (MIN.),ABORT-CURRENT,MIN BEAM (U),MAX. BEAM (U),DEFAULT CURRENT,DEFAULT BEAM (U),FIL. INCREASE
/BLOCK
12773   DATA MIN. #BLOCKS,MAX. #BLOCKS,MAX SIGMAMEAN(%),#SETS/BLOCK,MAX. GROWTH (%/MIN),PREHEAT CF (A),PREHEAT SF (A),NORMSPIKE#,""
12776!
12779   RESTORE 12767
12782   FOR J=1 TO 26
12785     READ A$
12788     PRINT J;TAB(6);A$[1,19];TAB(30);
12791     FOR Run=1 TO 16
12794       I=Run+16*Group
12797       V=RunvarO(I,J)
12800       IF NOT Std AND Run_order(I) THEN
12803         IF ((V=0 AND (J)1 AND J(5))) OR (J=24 AND V=0) OR (J=25 AND V=0) OR (J=26 AND V=0) OR (J=23 AND V=100) OR (Ogas(I) AN
D (J=5 OR J)11)) THEN
12806           PRINT " --";TAB(30+6*Run);
12809         ELSE
12812           PRINT V;TAB(30+6*Run);
12815         END IF
12818       END IF
12821       IF Std AND Stdvar(I,1) THEN
12824         V=Stdvar(I,J)
12827         IF ((V=0 AND (J)1 AND J(5))) OR (J=24 AND V=0) OR (J=25 AND V=0) OR (J=26 AND V=0) OR (J=23 AND V=100) THEN
12830           PRINT " --";TAB(30+6*Run);
12833         ELSE
12836           PRINT V;TAB(30+6*Run);
12839         END IF
12842       END IF
12845     NEXT Run
12848     PRINT
12851   NEXT J
12854   PRINT
12857   IF NOT Std THEN
12860     FOR R=1 TO 16
12863       Run=R+16*Group
12866       IF Run_order(Run) THEN PRINT TAB(24+6*R);Run;
12869     NEXT R
12872     PRINT USING "/,30X,K,#";"RUN-NUMBER"
12875   END IF
12878   PRINT USING "2/"
12881 NEXT Group
12884!
12887 PRINT USING "8/,2(K)";CHR$(27),"&k0S",CHR$(27),"&18D" ! Resume std typesize
12890 PRINTER IS CRT
12893 RETURN
12896!
12899 Store_runvars:! store the run-variables on  disk
12902 ON ERROR GOTO 12935
12905 ASSIGN @Path1 TO "RUNVAR:INTERNAL,4,1"
12908 GOTO 12917
12911! ON ERROR GOTO 6770
12914! ASSIGN @Path1 TO "RUNVAR:INTERNAL"
12917 ON ERROR GOTO 12935
12920 OUTPUT @Path1;RunvarO(*),Run_order(*),Run_name$(*),Runtype$(*),Run_iso(*)
12923 OFF ERROR
12926 ASSIGN @Path1 TO *
12929 RETURN
12932!
12935 OFF ERROR
```

```
12938  IF ERRN=59 THEN RETURN  !EOF- no data in file yet
12941  CALL Bad_vdisk(Std,1)
12944  GOTO Again
12947!
12950! When storing/retrieving runvars from disk, always try left-hand drive first, then right-hand drive if left doesn't work
12953!
12956 Store_stdvars:! store the std-run variables on left-hand disk
12959  ON ERROR GOTO 12935
12962  ASSIGN @Path1 TO "STDVAR:INTERNAL,4,1"
12965  GOTO 12974
12968! ON ERROR GOTO 6770
12971! ASSIGN @Path1 TO "STDVAR:INTERNAL"
12974  ON ERROR GOTO 12935
12977  OUTPUT @Path1;Stdvar(*),Std_iso(*),Std_name$(*),Stdtype$(*)
12980  OFF ERROR
12983  ASSIGN @Path1 TO *
12986  RETURN
12989!
12992 Get_stdvars:ON ERROR GOTO 13070! get the std-run variables from left disk
12995  DISP "Loading Std-Run Variables from disk..."
12998  ASSIGN @Path1 TO "STDVAR:INTERNAL,4,1"
13001  GOTO 13010
13004! ON ERROR GOTO 6809
13007! ASSIGN @Path1 TO "STDVAR:INTERNAL"
13010  ON ERROR GOTO 13070
13013  ENTER @Path1;Stdvar(*),Std_iso(*),Std_name$(*),Stdtype$(*)
13016  OFF ERROR
13019  Got_stdvars=1
13022  DISP
13025  RETURN
13028!
13031 Get_runvars:ON ERROR GOTO 13073! get run variables from left-hand disk
13034  DISP "Loading Run Variables from disk..."
13037  ASSIGN @Path1 TO "RUNVAR:INTERNAL,4,1"
13040  GOTO 13049
13043! ON ERROR GOTO 6809
13046! ASSIGN @Path1 TO "RUNVAR:INTERNAL"
13049  ON ERROR GOTO 13070
13052  ENTER @Path1;Runvar0(*),Run_order(*),Run_name$(*),Runtype$(*),Run_iso(*)
13055  OFF ERROR
13058  Got_runvars=1
13061  DISP
13064  RETURN
13067!
13070  OFF ERROR
13073  IF ERRN=59 THEN RETURN  !EOF- no data in file yet
13076  CALL Bad_vdisk(Std,0)
13079  GOTO Again
13082!
13085 Extract_stdvars:Std_name$(Run)=Response$(0) ! convert to MAIN format
13088  Stdtype$(Run)=Response$(2)
13091  FOR J=1 TO 8
13094    Std_iso(Run,J)=Isotope(J)
13097  NEXT J
13100  FOR V=1 TO 27
13103    Stdvar(Run,V)=Nresp(V)
13106  NEXT V
13109  Stdchange=1
13112  RETURN
13115!
```

```
13118 Extract_runvars: ! convert to MAIN program format
13121  Run_order(Run)=Nresp(-3)
13124  Runtype$(Run)=Response$(-2)
13127  Run_name$(Run)=Response$(0)
13130  FOR J=1 TO 8
13133    Run_iso(Run,J)=Isotope(J)
13136  NEXT J
13139  FOR V=1 TO 27   ! extract the 27 run-variables
13142    Runvar0(Run,V)=Nresp(V)
13145  NEXT V
13148  IF Run_order(Run) AND Run_name$(Run)<>"" AND Run_name$(Run)<>"***" THEN Sample_name$(Run_order(Run))=Run_name$(Run)
13151  Runchange=1
13154  RETURN
13157!
13160 Convert_stdvars:Lastresponse$(0)=Std_name$(Run) ! convert for MENU2 format
13163  Lastresponse$(-2)=Stdtype$(Run)
13166  Lastresponse$(-1)=VAL$(Std_iso(Run,1))
13169  FOR J=2 TO 8
13172    IF Std_iso(Run,J) THEN Lastresponse$(-1)=Lastresponse$(-1)&","&VAL$(Std_iso(Run,J))
13175  NEXT J
13178  FOR V=1 TO 27
13181    Lastresponse$(V)=VAL$(Stdvar(Run,V))
13184  NEXT V
13187  RETURN
13190!
13193 Convert_runvars: ! convert to MENU2  format
13196  Lastresponse$(0)=Run_name$(Run)
13199  Lastresponse$(-3)=VAL$(Run_order(Run))
13202  Lastresponse$(-2)=Runtype$(Run)
13205  Lastresponse$(-1)=VAL$(Run_iso(Run,1))
13208  FOR J=2 TO 8
13211    IF Run_iso(Run,J) THEN Lastresponse$(-1)=Lastresponse$(-1)&","&VAL$(Run_iso(Run,J))
13214  NEXT J
13217  FOR V=1 TO 27   ! extract the 27 run-variables
13220    Lastresponse$(V)=VAL$(Runvar0(Run,V))
13223  NEXT V
13226  RETURN
13229 SUBEND ! ----------------------------------------------------------------
13232 !
13235 !
13238 Beep:SUB Beep(Hertz,Seconds,Wait,Repeat,OPTIONAL Tune(*))
13241! if NPAR=4 then repeat a beep specified by the 1st 3 parameters.
13244! if NPAR=5 then beep a tone-pattern described by the Tune array.
13247  IF NPAR=4 THEN
13250    FOR I=1 TO Repeat
13253      BEEP Hertz,Seconds
13256      WAIT Wait
13259    NEXT I
13262    SUBEXIT
13265  ELSE
13268    I=1
13271    WHILE Tune(I,1)>0
13274      BEEP Tune(I,1),Tune(I,2)
13277      I=1+I
13280    END WHILE
13283  END IF
13286 SUBEND! ----------------------------------------------------------------
13289!
13292!
13295 Size:SUB Size(REAL Bmin,Bmax,Default_beam,Default_curr,Abort,INTEGER F,REAL Rate,Wait)
```

```
13298! adjust filament (at rate of Rate mA/sec to keep beamsize < Bmax U & > Bmin U (fil.curr. < Default_curr A), while allowing
13301! no more than Abort amps fil-curr.  F indicates center(1) or side (2) filaments.
13304  OPTION BASE 1
13307  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
13310  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
13313  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
13316  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
13319  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
13322  INTEGER Pr
13325  OFF KNOB
13328 !
13331 ! SUBFLAG key:  0 = OK;    1 = beam too large, so reduced current
13334 !               2 = beam too small and beyond abort-current
13337 !               3 = failed ENTER_BEAM check more than 6 times
13340 !               4 = failed filament flag-check in CHANGE_CURRENT > 3 times
13343 !
13346  FOR I=Auto*full_auto*2 TO 19
13349    ON KEY I LABEL "" CALL Clunk
13352  NEXT I
13355  ON KEY 9 LABEL "    ESCAPE" GOTO Exit_size
13358  Subflag=0
13361  Correct(Ff$(*),Mm$(*),Filament(*),Magnet(L,2),Mu,1,Coarsemag(L<>0),Foc(*))
13364  WAIT 1
13367  Screwy=0
13370  !
13373  REPEAT
13376    Enter_beam(X,Mu,L,1,Pr)
13379    Screwy=1+Screwy
13382  UNTIL Pr=1 OR Screwy>6
13385  !
13388  IF Screwy>6 THEN
13391    Subflag=3
13394    Error_message(Prtr(*),3,"FAILED ENTER_BEAM SUBFLAGS >6 TIMES IN SIZE")
13397    SUBEXIT
13400  END IF
13403  !
13406  Volts_beam=Mu/1000    ! VOLTS of beam
13409  IF Mu=0 AND Daly AND Mu<3 THEN
13412    Mu=1
13415    GOTO 13361
13418  END IF
13421  Bw=Bmax/(Bmin+1.E-9) ! size of beam-window, in ratio of max-beam to min-beam
13424  IF Volts_beam>=Bmin AND Volts_beam<=Bmax THEN SUBEXIT  ! beam within window
13427  IF Volts_beam>Bmax THEN  ! beam too large: reduce filament-current
13430    Subflag=1
13433    Pstep=.023-.005*(Bw<3)-.005*(Bw<2)-.005*(Bw<1.5) ! decrease filament-current by 2.3% if beam-window ratio>3, by 1.8% if <3,
>2, by 1.3% if <2,>1.5,
13436  ! and by 0.8% if <1.5
13439    Fstep=MAX(Pstep*Filament(F),.012) ! but not less than .012 amps
13442    Target=Filament(F)-Fstep
13445    GOTO Change_current
13448  END IF
13451  IF F=1 AND Nfils=2 THEN Increase_curr ! if center-fil beam for a triple
13454  IF Volts_beam<Default_beam THEN Increase_curr      ! beam < default-beam
13457  IF Filament(F)<Default_curr THEN
13460    GOTO Increase_curr                ! fil-curr<default curr
13463  ELSE ! beam less than Minbeam but greater than Defaultbeam, and fil-curr
13466      ! greater than default-current.
13469    SUBEXIT
13472  END IF
```

```
13475 !
13478 Increase_curr: !  beam too small- increase filament-current.
13481  IF Filament(F))=Abort THEN
13484    Subflag=2  ! beam too small, but fil-current ) abort current so stop trying
13487    GOTO Exit_size
13490  END IF
13493  Pstep=.019-.005*(Bw(3)-.005*(Bw(2)-.004*(Bw(1.5)  ! increase filament-current by 1.9% if beam-window ratio )3, by 1.4% if (2
)3,
13496 ! by 0.9% if )1.5 (2, and by 0.5% if (1.5.
13499  Fstep=MAX(Pstep*Filament(F),.009)  ! but not (.009 amps
13502  Target=Filament(F)+Fstep
13505  Bad_contacts=0
13508  !
13511 Change_current:REPEAT
13514    CALL Filament(10*(Target(=Filament(F))+Rate*(Target)Filament(F)),(Target),1,F)
13517    Bad_contacts=Bad_contacts+Subflag
13520  UNTIL Subflag=0 OR Bad_contacts)2
13523   !
13526  IF Bad_contacts)2 THEN
13529    Subflag=4
13532    Error_message(Prtr(*),3,"FAILED CONTACT-TEST 3 TIMES IN SIZE-INVOKED CHANGE_CURRENT")
13535    SUBEXIT
13538  END IF
13541  Wait(TIMEDATE,(Wait),Filament(F),Magnet(L,1),Auto,Full_auto)
13544  GOTO 13361
13547   !
13550 Exit_size:DISP
13553 SUBEND ! ------------------------------------------------------------
13556!
13559!
13562 Spikecorr:SUB Spikecorr(Aver(*),Last_aver(*),Ratio$(*),Acc(*),Lacc(*),Last_ratio$(*),Delta$(*),INTEGER Prtr(*),N,Block)
13565 ! correct observed ratios for spike isotopes and (linear) fractionation.  Dodson's algorithm.
13568  OPTION BASE 1
13571  COM /Spikedrun/ Spike$,Spikedrun_ratio$(*),Spike_ratio(*),Natural_ratio(*),INTEGER Iso_dif(*),Spkdrun_iso(*),Spkdrun_ref
13574  DIM Fract(2),A1(2),A2(2),G1(2),G2(2),Ca(2),Covrf(2),Srad(2),U(2),Sf(2),Spkdrun_sigma(3),Spkdrun_ratio(3),Rad(2)
13577  A=0
13580  B=0
13583  FOR I=1 TO 3-(Spkdrun_iso(3)=0 OR N=3)  ! find largest spike-ratio
13586    IF Spike_ratio(I))=B THEN
13589      B=Spike_ratio(I)
13592      A=I
13595    END IF
13598  NEXT I
13601  Smax=(Spike_ratio(A))1)*Spkdrun_iso(A)+(Spike_ratio(A)(=1)*Spkdrun_ref ! most abundant spike isotope
13604  Smult=(Spike_ratio(A))1)*Spike_ratio(A)+(Spike_ratio(A)(=1)  !Smax/reference isotope for spike
13607  MAT Spkdrun_ratio= (0)
13610  FOR K=1 TO 3-(Spkdrun_iso(3)=0 OR N=3)
13613    FOR L=1 TO N-1
13616      IF (NOT POS(Ratio$(L),VAL$(Spkdrun_iso(K)))) OR (NOT POS(Ratio$(L),VAL$(Spkdrun_ref))) THEN 13625
13619      Spkdrun_ratio(K)=Aver(L)^(1-2*(VAL(Ratio$(L))=Spkdrun_ref)) ! spiked-run obs. ratios
13622      Spkdrun_sigma(K)=Acc(L)*Spkdrun_ratio(K)/100                ! ''  ''  std devs
13625    NEXT L
13628    IF Spkdrun_ratio(K)=0 THEN SUBEXIT
13631  NEXT K
13634  N=N+1+(N=4) ! add "SAM/SPK" ratio & XXX*XXX ratio (radiogenic)
13637              ! (N passed in parens, so shouldn't affect N in main program
13640  K3=Iso_dif(1)*(Natural_ratio(2)/Spkdrun_ratio(2)-1)-Iso_dif(2)*(Natural_ratio(1)/Spkdrun_ratio(1)-1)
13643  K4=Iso_dif(2)*(Spike_ratio(1)/Spkdrun_ratio(1)-1)-Iso_dif(1)*(Spike_ratio(2)/Spkdrun_ratio(2)-1)
13646  R=K3/K4 ! ratio of spike reference-isotope to sample reference-isotope.
13649  FOR I=1 TO 2
```

```
13652   Za(I)=Natural_ratio(I)+R*Spike_ratio(I)
13655  NEXT I
13658  Ca(1)=Iso_dif(2)*Za(1)/(K4*Spkdrun_ratio(1)^2)
13661  Ca(2)=Iso_dif(1)*Za(2)/(K4*Spkdrun_ratio(2)^2)
13664  Sr=SQR(ABS(Ca(1)^2*Spkdrun_sigma(1)^2+Ca(2)^2*Spkdrun_sigma(2)^2)) ! std deviation of R
13667  IF N=6 THEN A3=Iso_dif(3)*Spkdrun_ratio(3)*(1+R)
13670  FOR I=1 TO 2
13673   Fract(I)=Za(I)/(Iso_dif(I)*Spkdrun_ratio(I)*(1+R))-1/Iso_dif(I) ! fractionation calculated from Spkdrun_ratio(I)
13676   U(I)=1+Iso_dif(3)*Fract(I)
13679   A1(I)=(1+R)*U(I)
13682   A2(I)=Spkdrun_ratio(3)*U(I)-Spike_ratio(3)
13685   G1(I)=(Spike_ratio(I)-Natural_ratio(I))/(Iso_dif(I)*Spkdrun_ratio(I)*(1+R)^2)
13688   G2(I)=Za(I)/(Iso_dif(I)*Spkdrun_ratio(I)^2*(1+R))
13691   Psi=(I=1)-(I=2)
13694   Sf(I)=SQR(G1(I)^2*Sr^2+G2(I)^2*Spkdrun_sigma(I)^2-2*Psi*G1(I)*G2(I)*Ca(I)*Spkdrun_sigma(I)^2) ! std dev. of fractionation f
rom  Spkdrun_ratio(I)
13697   Covrf(I)=G1(I)*Sr^2-Psi*Ca(I)*G2(I)*Spkdrun_sigma(I)^2 !COV(R,Fi)
13700   IF N=6 THEN Srad(I)=SQR(ABS(A1(I)^2*Spkdrun_sigma(3)^2+Iso_dif(2)^2*Sr^2+A3^2*Sf(I)^2+2*A2(I)*A3*Covrf(I)))
13703   ! std. dev. of radiogenic ratio from Spkdrun_ratio(I)
13706  NEXT I
13709  Q=1+(Srad(2)<Srad(1)) ! Spkdrun_ratio(I) with lowest uncertainty
13712  IF N=6 THEN
13715   FOR I=1 TO 2
13718    Rad(I)=(1+R)*U(I)*Spkdrun_ratio(3)-R*Spike_ratio(3)
13721   NEXT I
13724   Aver(4)=(Rad(1)/Srad(1)^2+Rad(2)/Srad(2)^2)/(1/Srad(1)^2+1/Srad(2)^2)  ! spike & fractionation-corrected radiogenic-isotope
 ratio
13727    ! (wted average of value from both nonrad-ratios)
13730   Acc(4)=100*Srad(Q)/Aver(N-2)
13733   Ratio$(4)=Spikedrun_ratio$(3)
13736   IF Ratio$(4)=Last_ratio$(4) AND Block>1 THEN CALL Delta(Delta$(4),Aver(4),Last_aver(4),Acc(4),Last_acc(4)) ! e.g. 143*144,
87*86
13739  END IF
13742  Aver(N-1)=1/(R*Smult) ! ratio of sample reference-isotope to most-abundant spike-isotope
13745  Acc(N-1)=100*Sr/R
13748  Ratio$(N-1)="SAM/SPK"
13751  IF Ratio$(N-1)=Last_ratio$(N-1) AND Block>1 THEN CALL Delta(Delta$(N-1),Aver(N-1),Last_aver(N-1),Acc(N-1),Lacc(N-1))
13754  FOR I=2 TO 1 STEP -1
13757   PRINTER IS Prtr(I)
13760   IF I=2 THEN PRINT USING "#,/,K";CHR$(27)&"&l6D" ! 6 lines/inch (wide)
13763   PRINT "SAMPLE-"&VAL$(Spkdrun_ref)&"/SPIKE-"&VAL$(Smax)&" =";DROUND(Aver(N-1),5);"+/-";VAL$(ABS(DROUND(Acc(N-1),2)))&"%";
13766   IF Delta$(N-1)<>"" THEN PRINT TAB(60);"[delta= "&TRIM$(Delta$(N-1))&"%]";
13769   PRINT
13772   PRINT "FRACTIONATION= ";
13775   IF FRACT(Q)>1 THEN PRINT "+";
13778   PRINT VAL$(DROUND(100*Fract(Q),3));" %/a.m.u.          (Spiked with ";TRIM$(Spike$);")"
13781   IF N=6 THEN
13784    PRINT "CORRECTED "&VAL$(Spkdrun_iso(3))&"/"&VAL$(Spkdrun_ref)&" =";DROUND(Aver(4),6);"+/-";ABS(DROUND(Acc(4),2));"%";
13787    IF Delta$(N-2)<>"" THEN
13790     PRINT TAB(53);"[delta= "&TRIM$(Delta$(N-2))&"%]"
13793    ELSE
13796     PRINT
13799    END IF
13802   END IF
13805   Last_ratio$(N-1)=Ratio$(N-1)
13808   IF N=6 THEN Last_ratio$(4)=Ratio$(4)
13811   PRINT USING "K,/";RPT$("-",80)
13814   IF I=2 THEN PRINT USING "#,K";CHR$(27)&"&l8D"
13817  NEXT I
13820 SUBEND ! ----------------------------------------------------------------
```

```
13823 !
13826  !
13829 Dispikes:SUB Dispikes ! show #s & names of spikes defined on disk
13832  OPTION BASE 1
13835  COM /Spikedrun/ Spike$,Spikedrun_ratio$(*),Spike_ratio(*),Natural_ratio(*),INTEGER Iso_dif(*),Spikedrun_iso(*),Spkdrun_ref
13838  DIM Drive$(0:1)[5]
13841  Drive$(1)="LEFT"
13844  Drive$(0)="RIGHT"
13847  OUTPUT KBD;CHR$(255)&CHR$(75);
13850  FOR Disk=1 TO 0 STEP -1
13853    ON ERROR GOTO 13865
13856    ASSIGN @Path1 TO "SPIKE:INTERNAL,4,"&VAL$(Disk)
13859    PRINT "SPIKES DEFINED ON DISK IN "&Drive$(Disk)&" DRIVE:"
13862    GOTO 13877
13865  NEXT Disk
13868  Clunk
13871  PRINT FNH$(" NO SPIKE FILE PRESENT IN EITHER DRIVE ")
13874  SUBEXIT
13877  PRINT USING "/,8A,8X,6A,7X,5A,/";FNUn$("NUMBER"),FNUn$("NAME"),FNUn$("REF")
13880  ON ERROR GOTO 13913
13883  FOR I=1 TO 20
13886    ENTER @Path1,I;Spike$,Spkdrun_ref,Spikedrun_iso(*),Spikedrun_ratio$(*),Iso_dif(*),Spike_ratio(*),Natural_ratio(*)
13889    IF TRIM$(Spike$)<>"" THEN
13892      PRINT " ";I;"------- ";Spike$;TAB(25);Spkdrun_ref;TAB(31);
13895      FOR J=1 TO 3
13898        IF Spikedrun_iso(J) THEN PRINT Spikedrun_iso(J);TAB(31+5*J);
13901      NEXT J
13904      PRINT
13907    END IF
13910  NEXT I
13913  PRINT
13916  OFF ERROR
13919 SUBEND ! ----------------------------------------------------------------
13922  !
13925  !
13928 Cubic:SUB Cubic(Coef(*),Peak_t(*),Peak_array(*),INTEGER Nsets,I)
13931  ! calculate least-squares cubic fit to monitor isotope with time
13934  OPTION BASE 1
13937  DIM Time_array(40,4),Peak(40,1),Xa(4,40),Xb(4,4),Xc(4,4),Xd(4,40)
13940  REDIM Time_array(Nsets,4),Peak(Nsets,1),Xa(4,Nsets),Xd(4,Nsets)
13943  FOR J=1 TO Nsets
13946    Time_array(J,1)=1 ! power-array of X-values (1,X,X^2,X^3)
13949    FOR P=2 TO 4
13952      Time_array(J,P)=Time_array(J,P-1)*Peak_t(I,J)
13955    NEXT P
13958    Peak(J,1)=Peak_array(I,J)  ! vector of Y-values
13961  NEXT J
13964  MAT Xa= TRN(Time_array)
13967  MAT Xb= Xa*Time_array
13970  MAT Xc= INV(Xb)
13973  MAT Xd= Xc*Xa
13976  MAT Coef= Xd*Peak    ! coefficents of best-fit 3rd-order polynomial
13979 SUBEND ! ----------------------------------------------------------------
13982 !
13985 Focscan:SUB Focscan(INTEGER Foc(*),I_t) ! scan focus-potentials
13988  OPTION BASE 1
13991  DIM Prompt$(4)[36],Response$(4)[50],Use(4),Range(4,2)
13994  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
13997  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
14000  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
```

```
14003  INTEGER Pr
14006  Z$="123456789"
14009  OFF KNOB
14012  ON KEY 9 LABEL "    ESCAPE" GOTO 14114
14015  PRINT USING "K,2/,8(3D,3X),/,8(4D,2X),/";"FOCUSING VALUES:",1,2,3,4,5,6,7,8,Foc(*)
14018  DATA Plate# (1-8),Min. Value,Max. Value,Scan-Step
14021  DATA "1","0","999","1",1,8,0,998,1,999,1,100
14024  READ Prompt$(*),Response$(*),Range(*)
14027  IF Nfils=1 THEN  ! don't allow plate-8 scan if not a triple
14030    Range(1,2)=7
14033    Prompt$(1)[11,11]="7"
14036  END IF
14039  MAT Use= (1)
14042  Form(Prompt$(*),Response$(*),Use(*),Range(*),4,1,E)
14045  IF E THEN SUBEXIT
14048  OUTPUT KBD;CHR$(255)&CHR$(75);
14051  Plate=VAL(Response$(1))
14054  Xmin=VAL(Response$(2))
14057  Xmax=VAL(Response$(3))
14060  Step=VAL(Response$(4))
14063  IF (Plate=8) AND (Xmin<450) THEN Xmin=450
14066  IF (Plate=8) AND (Xmax>550) THEN Xmax=550
14069  IF Step<1 THEN Step=1
14072  Enter_beam(X,Mv,1,I_t,Pr)
14075  IF Pr>1 THEN 14018
14078  ON KEY 3 LABEL " DOUBLE STEP" GOSUB Double
14081  ON KEY 8 LABEL "  HALVE STEP" GOSUB Halve
14084  ON KEY 9 LABEL "    ESCAPE" GOTO 14114
14087  Axes(0,100,8,100,Xmin,Xmax,0,1.3*Mv,"PLATE "&VAL$(Plate),"mV BEAM",0,Mu,0)
14090  I=Xmin
14093  WHILE I<Xmax
14096    OUTPUT 8 USING "2(4A,4Z)";"$0F"&Z$[10-Plate,10-Plate],I,"$0MV",I
14099    Enter_beam(X,Mv,1,I_t,Pr)
14102    IF I>Xmin THEN PLOT I,Mv
14105    IF I=Xmin THEN MOVE I,Mv
14108    I=I+Step
14111  END WHILE
14114  OUTPUT 8 USING "4A,4Z";"$0F"&Z$[10-Plate,10-Plate],Foc(Plate)
14117  WAIT 2
14120  GRAPHICS OFF
14123  ALPHA ON
14126  GOTO 14042
14129 Double:Step=2*Step
14132  BEEP 880,.07
14135  RETURN
14138 Halve:IF Step>=2 THEN Step=Step/2
14141  BEEP 220,.07
14144  RETURN
14147 SUBEND ! ------------------------------------------------------------
14150  !
14153  !
14156 Calibrate:SUB Calibrate(REAL Filament(*),INTEGER 8,Peak_inter,Daly,I_t,Mu,Mm$(*),Collector$(*))      ! Calibrate Daly gain
14159  OPTION BASE 1
14162  COM /Specs/ Mx(*),Ions,Ze(*),Noise(*)
14165  DIM Mv_beam(3),New_mx(0:1)
14168  OFF KEY
14171  OFF KNOB
14174  ON KEY 9 LABEL "    ESCAPE" GOTO 14285
14177  Mu=0
14180  I_t=1
```

```
14183  MAT New_mx= Mx
14186  OUTPUT KBD;CHR$(255)&CHR$(75);
14189  OUTPUT 8;"$IDV"
14192  ENTER 8;Count
14195  Beamsize=(Count-Ze(0))/New_mx(0)
14198  IF (Beamsize(1) OR (Beamsize>50) OR NOT Daly THEN
14201    PRINT USING "3/,K,/";FNH$("BEAM MUST BE >1 mV  AND  <50 mV, DALY")&FNB1$("ENABLED")&FNH$(", TO CALIBRATE DALY GAIN.")
14204    Clunk
14207    GOTO 14285
14210  END IF
14213  OUTPUT 8 USING "4A,42";"$OFJ",B-Peak_inter/2  ! set magnet to -1/2 mass offset for zero-calibration
14216  Zero(Filament(*),0,Noise_time,Daly,B,Peak_inter,Mu,Mm$(*),Collector$(*))
14219  OUTPUT KBD;CHR$(255)&CHR$(75);
14222  PRINT USING "K,2/";FNUn$("CALIBRATION OF DALY GAIN")
14225  OUTPUT 8 USING "4A,42";"$OFJ",B
14228  FOR I=1 TO 3
14231    Mu=(I=2)
14234    OUTPUT 8;Mm$(Mu,1)
14237    WAIT (2+5*(I<>2))
14240    Seconds_integr=10*(1+(NOT Mu)*((Beamsize(5)+(Beamsize(2)))
14243  !  10-sec. integration for Daly, 10 sec. for Cup if >5mV, 20 sec if 2-5mV, 30 sec if 1-2 mV
14246    FOR J=1 TO Seconds_integr
14249      OUTPUT 8;"$IDV"
14252      ENTER 8;Beamcount
14255      Beam_mv=(Beamcount-Ze(Mu))/New_mx(Mu)
14258      Mv_beam(I)=Mv_beam(I)+Beam_mv/Seconds_integr
14261      DISP Collector$(Mu),J;DROUND(Beam_mv,3)
14264    NEXT J
14267    PRINT Collector$(Mu);TAB(12);DROUND(Mv_beam(I),4)
14270  NEXT I
14273  Oldgain=New_mx(1)/New_mx(0)
14276  New_mx(1)=New_mx(1)*Mv_beam(2)/((Mv_beam(1)+Mv_beam(3))/2)
14279  PRINT USING "2/,5(K),2/";"DALY GAIN IS ",DROUND(New_mx(1)/New_mx(0),3),"  (LAST CALIBRATION WAS ",DROUND(Oldgain,3),")"
14282  MAT Mx= New_mx
14285  I_t=2
14288  OUTPUT 8;Mm$(Mu,I_t)
14291  BEEP
14294 SUBEND ! ------------------------------------------------------------
14297 !
14300 !
14303 Zero:SUB Zero(REAL Filament(*),Zero_time,Noise_time,INTEGER Daly,B,Peak_inter,Mu,Mm$(*),Collector$(*))
14306 ! take collector zeroes (no more than every hour) and measure dark-noise (no more than every 6 hours)
14309  OPTION BASE 1
14312  COM /Specs/ Mx(*),Ions,Ze(*),Noise(*)
14315  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),Itime
14318  DIM Sums(5),Reading(20),Count_noise(0:1),New_zeroes(0:1),New_noise(0:1)
14321  INTEGER Count_time
14324  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
14327  MAT New_zeroes= Ze
14330  MAT New_noise= Noise
14333  OFF KEY
14336  OFF KNOB
14339  ON KEY 9 LABEL "    ESCAPE" GOTO 14663
14342  GRAPHICS OFF
14345  Subflag=0
14348  IF New_zeroes(1)=0 AND Daly THEN
14351    Zero_time=0
14354    Noise_time=0
14357  END IF
14360  IF (TIMEDATE-Zero_time)/3600<1 THEN SUBEXIT ! zeros taken within last hour
```

```
14363  PRINT USING "3/"
14366  PRINT USING "K,2/";FNUn$("  COLLECTOR ZERO & NOISE CALIBRATION: ")&"      Please Wait..."
14369  Nocurrent=(Filament(1)=0) AND (Filament(2)=0)
14372  OUTPUT 8;Mm$(0,2-Nocurrent)
14375  MAT New_zeroes= (0)
14378  M=Mu
14381  WAIT (1+5*M)
14384  IF NOT Nocurrent THEN
14387    !
14390  ! Scan magnet for lowest zero if filaments have currents
14393    GOSUB Scanfor_min
14396    IF Max_count<40000 THEN
14399      OUTPUT 8;Mm$(1,1)
14402      Mu=1
14405      GOSUB Scanfor_min
14408      GOTO 14507
14411    ELSE
14414      PRINT USING "10/,K";"CAN'T FIND A GOOD ZERO-POSITION"
14417      Clunk
14420      GOSUB Restore
14423      SUBEXIT
14426    END IF
14429  !
14432 Scanfor_min:Zero_location=300
14435    OUTPUT 8 USING "4A,4Z";"$OFJ",Zero_location-7*Peak_inter/8
14438    WAIT 2
14441    Min_count=1.E+99
14444    Max_count=0
14447    FOR C=7 TO 0 STEP -1    ! find magnet-setting giving lowest zero
14450      Mag_pos=INT(Zero_location-C*Peak_inter/8)
14453      IF Mag_pos>0 AND Mag_pos<1.E+4 THEN
14456        OUTPUT 8 USING "4A,4Z";"$OFJ",Mag_pos
14459        OUTPUT 8;"$IDV"
14462        ENTER 8;Count
14465        IF Mu=0 THEN DISP "Looking for best zero-position (CUP) ";Mag_pos,Count
14468        IF Mu=1 THEN DISP "Looking for best zero-position (DALY) ";Mag_pos,Count
14471        IF Count<Min_count THEN
14474          Min_count=Count
14477          Mag_min=Mag_pos
14480        END IF
14483        IF Count>Max_count THEN Max_count=Count
14486      END IF
14489    NEXT C
14492    OUTPUT 8 USING "4A,4Z";"$OFJ",Mag_min
14495    OUTPUT 8;Mm$(0,1)
14498    RETURN
14501  END IF
14504    !
14507  Hours=(TIMEDATE-Noise_time)/3600 ! hours since last collector-noise measurement
14510  IF Hours>6 THEN Check_noise=1
14513  FOR M=0 TO Daly>0
14516    OUTPUT 8;Mm$(M,1)
14519    Count_time=14-4*M+6*Check_noise
14522    MAT Sums= (0)
14525    FOR Ct=-5 TO Count_time
14528      OUTPUT 8;"$IDV"
14531      ENTER 8;Count
14534      DISP Collector$(M);"  ZEROES";TAB(18);CHR$(128+(Ct>0));Count+1;Cn$,TAB(40);Count_time-Ct+1
14537      IF Ct>0 THEN
14540        Reading(Ct)=Count
```

```
14543       CALL Sums(Sums(*),1,0,Count,0)
14546     END IF
14549   NEXT Ct
14552   Calc_sigma(Count_noise(M),Sums(3),Sums(4),Count_time)
14555   Est_zero=Sums(3)/Count_time
14558   FOR I=1 TO Count_time  ! 2-sigma rejection of outliers
14561     IF ABS(Est_zero-Reading(I))>2*Count_noise(M) THEN CALL Sums(Sums(*),-1,0,Reading(I),Count_time)
14564   NEXT I
14567   Calc_sigma(Count_noise(M),Sums(3),Sums(4),Count_time)
14570   IF Check_noise THEN New_noise(M)=Count_noise(M)/Mx(M)  ! noise in mV
14573   New_zeroes(M)=Sums(3)/Count_time
14576 NEXT M
14579 PRINT USING "33X,7A,9X,8A,/";FNUn$(" CUP "),FNUn$(" DALY ")
14582 PRINT FNH$("ZEROES:")&"       (counts/second)";TAB(36);INT(New_zeroes(0));TAB(51);INT(New_zeroes(1))
14585 IF Check_noise THEN PRINT FNH$("DARK-NOISE:")&"   (mV/second)";TAB(35);DROUND(New_noise(0),3);TAB(50);DROUND(New_noise(1),2);
CHR$(10)
14588 GOSUB Restore
14591 IF New_zeroes(0)>800 OR New_zeroes(0)<300 OR (Daly AND (New_zeroes(1)>900 OR New_zeroes(1)<300)) THEN Bad_readings
14594 IF Check_noise AND New_noise(0)>.1 OR New_noise(1)>.01 THEN Bad_readings
14597 IF Check_noise THEN Noise_time=TIMEDATE
14600 Zero_time=TIMEDATE
14603 MAT Ze= New_zeroes
14606 MAT Noise= New_noise
14609 SUBEXIT
14612 !
14615 Restore:OUTPUT 8;Mm$(Mu,Itime)      ! restore collector, integr. time
14618 OUTPUT 8 USING "4A,4Z";"OFJ",B     ! restore magnet-position
14621 RETURN
14624 !
14627 Bad_readings:Noise_time=0
14630 Subflag=1
14633 FOR P=1+Full_auto TO 1 STEP -1
14636   PRINTER IS Prtr(P)
14639   IF P=2 THEN PRINT USING "3/,80A,/,80A,/";RPT$("*",80),RPT$("*",80)
14642   PRINT USING "K,/,K";" ZERO(ES) MUST BE BETWEEN 400 AND 800 "," NOISE MUST BE <0.1 mV (CUP), <.01 mV (DALY) "
14645   PRINT USING "2/,K,/";"CHECK "&FNH$("GAIN")&" (x1 FOR CUP, 1E-05 Amps Full Scale FOR DALY)"
14648   PRINT FNH$("OFFSET")&" (ABOUT 5.50 FOR CUP), "&FNH$("ZERO")&" (DALY)"
14651   PRINT USING "3/,K,/";"RE-TAKE ZEROES BY PRESSING "&FNUn$("CTL")&" Z DURING BMC"
14654   IF P=2 THEN PRINT USING "/,80A,/,80A,4/";RPT$("*",80),RPT$("*",80)
14657 NEXT P
14660 Clunk
14663 SUBEND ! ---------------------------------------------------------------
14666 !
14669 Mfocus:SUB Mfocus(Stripchart,Logmon,Time0,INTEGER I_t,Foc(*),Z$) ! Manual focus-change
14672 OPTION BASE 1
14675 INTEGER Maxplate(8),Minplate(8),Pr
14678 DIM P$(8)[5]
14681 COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
14684 COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
14687 COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
14690 COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
14693 COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
14696 MAT Maxplate= (999)
14699 MAT Minplate= (0)
14702 Maxplate(8)=540
14705 Minplate(8)=460
14708 I_t=2
14711 OUTPUT 8;Mm$(Mu,I_t) ! .2 second integr-time
14714 OUTPUT KBD;Clear$;
14717 FOR I=1 TO 8
```

```
14720   P$(I)=FNH$(VAL$(I))
14723  NEXT I
14726  IF NOT Stripchart THEN
14729    PRINT USING "K,2/,K";"MANUAL-FOCUS ROUTINE: ","Press a softkey or a number-key to indicate which plate to focus,"
14732    PRINT USING "K,2/,K,/";"use "&FNH$("KNOB")&" or up/down arrows, or the + - keys to adjust setting of plate.","Press "&FNH$(
"k9")&" to escape."
14735    PRINT "(NOTE:  You can focus in this way but with real-time ion-beam graphics if"
14738    PRINT "a BEAM CHART ( k3  during BMC) was in use before the manual-focus request)"
14741  END IF
14744  A=KNOBX
14747  Plate=1
14750  P$(1)=FNB1$("1")
14753  I_t=2
14756  FOR I=1 TO 6+Nfils
14759    ON KEY I LABEL "    PLATE "&VAL$(I) GOTO 14792
14762  NEXT I
14765  ON KEY 9 LABEL "    ESCAPE" GOTO Exit
14768  IF Stripchart THEN
14771    GRAPHICS ON
14774  ELSE
14777    GOSUB Print_plates
14780    PRINT TABXY(1,16);"PLATE#"
14783    PRINT TABXY(1,17);"SETTING"
14786  END IF
14789  GOSUB Print_values
14792  GOTO 14852
14795   !
14798 Print_plates:FOR I=1 TO 6+Nfils
14801    IF NOT Stripchart THEN PRINT TABXY(11+(I-1)*7,16);I
14804  NEXT I
14807  RETURN
14810   !
14813  IMAGE "FOCUS:    1 ",3D,"    2 ",3D,"    3 ",3D,"    4 ",3D,"    5 ",3D,"    6 ",3D,"    7 ",3D,"    8 ",3D
14816  IMAGE "FOCUS:",8(3X,5A,3D)
14819 Print_values:IF NOT Stripchart THEN
14822    FOR I=1 TO 6+Nfils
14825      PRINT TABXY(10+(I-1)*7,17);Foc(I)
14828    NEXT I
14831  ELSE
14834    GOSUB Plates
14837  END IF
14840  RETURN
14843 Plates:DISP USING 14816;P$(1),Foc(1),P$(2),Foc(2),P$(3),Foc(3),P$(4),Foc(4),P$(5),Foc(5),P$(6),Foc(6),P$(7),Foc(7),P$(8),Foc(8
)
14846  RETURN
14849   !
14852  IF NOT Stripchart THEN GOSUB Intro
14855  Enter_beam(Counts,Mv,L,I_t,Pr)
14858  WAIT .10
14861  ON CYCLE .195 GOSUB Query_beam
14864  ON KBD ALL GOSUB Change_plate
14867  ON KNOB .10 GOSUB Change_setting
14870  GOTO 14870
14873   !
14876 Query_beam:Enter_beam(Counts,Mv,L,2,Pr)
14879  IF Pr=3 THEN Exit
14882  T=(TIMEDATE-Time0)/60
14885  IF Stripchart THEN
14888    IF NOT Logmon THEN DRAW T,Mv
14891    IF Logmon THEN DRAW T,LGT(ABS(Mv+(NOT Mv)))
```

```
14894  ELSE
14897    IMAGE 2X," mV",X,K," ",3X,K
14900    DISP USING 14897;Magnet(L,1),DROUND(Mv,4)
14903  END IF
14906  RETURN
14909 !
14912 Change_plate:CALL Kybrd(KBD$,K)
14915  SELECT K
14918  CASE 49 TO 54+Nfils        ! keys 1 thru 7 or 8 pressed
14921    Lastplate=Plate
14924    Plate=K-48
14927  CASE -207 TO -202+Nfils    ! keys k1 thru k7 or k8 pressed
14930    Lastplate=Plate
14933    Plate=K+208
14936  CASE -162,43               ! up-arrow or plus key pressed
14939    Change=1
14942    GOTO 15020
14945  CASE -170,45               ! down-arrow or minus key pressed
14948    Change=-1
14951    GOTO 15020
14954  CASE -199                  ! k9 pressed
14957    GOTO Exit
14960  CASE -176                  ! PAUSE key pressed
14963    PAUSE
14966  CASE ELSE
14969    Clunk
14972    RETURN
14975  END SELECT
14978  BEEP 440,.08
14981  IF NOT Stripchart THEN
14984    GOSUB Print_plates
14987 Intro:PRINT TABXY(11+(Plate-1)*7,16);Ci$;Plate;Cn$
14990  END IF
14993  IF Stripchart THEN
14996    P$(Lastplate)=FNH$(VAL$(Lastplate))
14999    P$(Plate)=FNB1$(VAL$(Plate))
15002    GOSUB Plates
15005  END IF
15008  Setting=Foc(Plate)
15011  RETURN
15014                !
15017 Change_setting:Change=INT(KNOBX/3)
15020  Setting=Foc(Plate)+Change
15023  IF Setting>=Minplate(Plate) AND Setting<=Maxplate(Plate) THEN 15050
15026  IF Setting<Minplate(Plate) THEN
15029    BEEP 100,.1
15032    Setting=Minplate(Plate)
15035  END IF
15038  IF Setting>Maxplate(Plate) THEN
15041    BEEP 2000,.1
15044    Setting=Maxplate(Plate)
15047  END IF
15050  Foc(Plate)=Setting
15053  IF NOT Stripchart THEN PRINT TABXY(10+(Plate-1)*7,17);Foc(Plate)
15056  IF Stripchart THEN GOSUB Print_values
15059  OUTPUT 8 USING "2(4A,4Z)";"$OF"&Z$[10-Plate,10-Plate],Setting,"$OMV",Setting
15062  RETURN
15065 Exit:OFF CYCLE
15068  OFF KNOB
15071  OFF KBD
```

```
15074 SUBEND ! ------------------------------------------------------------
15077 !
15080 !
15083 Nspike:SUB Nspike
15086  ! define values for a new spike
15089  OPTION BASE 1
15092  COM /Spikedrun/ Spike$,Spikedrun_ratio$(*),Spike_ratio(*),Natural_ratio(*),INTEGER Iso_dif(*),Spikedrun_iso(*),Spkdrun_ref
15095  DIM Response$(5)[50],Prompt$(5)[36],Use(5),Range(5,2)
15098  CALL Dispikes
15101  OFF KEY
15104  OFF KNOB
15107  PRINT TABXY(1,16);"ENTER NUMBER (1-20)  AND NAME  (must include element-symbol) OF NEW SPIKE:"
15110  PRINT TABXY(1,18);"(e.g. "&CHR$(34)&"5,Sr_KRL"&CHR$(34)&".   Enter 0,0 to return to BMC)"
15113  INPUT Spike,Spike$[1,8]
15116  IF Spike=0 THEN SUBEXIT
15119  IF Spike>20 THEN 15107
15122  REDIM Prompt$(4),Response$(4),Use(4),Range(4,2)
15125  DATA REFERENCE ISOTOPE (eg 86 for Sr),NONRADIOGENIC ISOT #1 (eg 84 for Sr),NONRADIOGENIC ISOT #2 (eg 88 for Sr),RADIOGENIC IS
OT if any; eg 87 for Sr
15128  DATA ??,??,??,"",10,300,10,300,10,300,-1,-1
15131  READ Prompt$(*),Response$(*),Range(*)
15134  MAT Use= (1)
15137  Form(Prompt$(*),Response$(*),Use(*),Range(*),4,1,E)
15140  IF E THEN SUBEXIT
15143  Spkdrun_ref=VAL(Response$(1))
15146  Spikedrun_iso(1)=VAL(Response$(2))
15149  Spikedrun_iso(2)=VAL(Response$(3))
15152  Spikedrun_iso(3)=0
15155  ON ERROR GOTO 15161
15158  Spikedrun_iso(3)=VAL(Response$(4))
15161  OFF ERROR
15164  FOR I=1 TO 3
15167    Spikedrun_ratio$(I)=VAL$(Spikedrun_iso(I))&"/"&VAL$(Spkdrun_ref)
15170  NEXT I
15173  REDIM Prompt$(5),Response$(5),Use(5),Range(5,2)
15176  Prompt$(1)=Spikedrun_ratio$(1)&" OF NATURAL ELEMENT"
15179  Prompt$(2)=Spikedrun_ratio$(2)&" OF NATURAL ELEMENT"
15182  Prompt$(3)=Spikedrun_ratio$(1)&" OF SPIKE"
15185  Prompt$(4)=Spikedrun_ratio$(2)&" OF SPIKE"
15188  IF Spikedrun_iso(3) THEN
15191    N=5
15194    Use(5)=1
15197    Prompt$(5)=Spikedrun_ratio$(3)&" OF SPIKE"
15200  ELSE
15203    N=4
15206    Use(5)=0
15209  END IF
15212  DATA 0,9E99,0,9E99,0,9E99,0,9E99,0,9E99
15215  READ Range(*)
15218  MAT Response$= ("??")
15221  Form(Prompt$(*),Response$(*),Use(*),Range(*),N,1,E)
15224  IF E THEN SUBEXIT
15227  Natural_ratio(1)=VAL(Response$(1))
15230  Natural_ratio(2)=VAL(Response$(2))
15233  Spike_ratio(1)=VAL(Response$(3))
15236  Spike_ratio(2)=VAL(Response$(4))
15239  IF N=5 THEN Spike_ratio(3)=VAL(Response$(5))
15242  FOR I=1 TO 3
15245    IF I<3 OR N=5 THEN Iso_dif(I)=Spikedrun_iso(I)-Spkdrun_ref
15248  NEXT I
```

```
15431   IF Peak(N)=0 THEN Scan_all=1 ! if no value for this peak, must scan all peaks.  Should prevent errors from early Data-exits
.
15434  NEXT I
15437  IF Daly_count<N THEN Mu=0 ! Stay on Daly if all peaks small enough
15440  Wait=1
15443  FOR I=1 TO N-1
15446   !  increase wait-time if change in isotope is more than 10%
15449    IF ABS(Magnet(Order(I),1)-Magnet(Order(I+1),1)))Magnet(Order(I),1)/10 THEN Wait=2
15452  NEXT I
15455  OUTPUT 8;Mm$(Mu,1)
15458  IF M>Mu THEN WAIT 6
15461  MAT Peak= (0)
15464  FOR J=0 TO 1
15467    FOR I=(NOT J)+N*J TO N*(NOT J)+J STEP (NOT J)-J  ! Step-scan up, down over isotopes
15470      IF Scan_all OR I=1 THEN
15473        GOSUB 15518
15476        ON Pr GOTO 15482,15443,15443
15479      END IF
15482    NEXT I
15485  NEXT J
15488  DISP
15491  Q=(MAX(Peak(*)))40)
15494  IF Q=0 AND Mu=0 AND Daly=1 THEN   ! use Daly for data-taking if all peaks <35 mV & Daly-variable=1
15497    Mu=1
15500    OUTPUT 8;Mm$(Mu,1)
15503    PRINT "ON DALY"
15506    GOTO 15461
15509  END IF
15512  Data_daly=Mu*(Daly=1)
15515  SUBEXIT
15518  Correct(Ff$(*),Mm$(*),Filament(*),Magnet(Order(I),2),Mu,1,Coarsemag(1),Foc(*))
15521  FOR K=1 TO Wait+1
15524    IF J*(K(=Wait)*(I=N) THEN 15539
15527    Enter_beam(0,Mv,Order(I),1,Pr)
15530    IF Pr>1 THEN Rough
15533    DISP "QUICK SCAN:   ";Ci$;Magnet(Order(I),1);Cn$;TAB(25);DROUND(Mv,3);"mV"
15536    IF K>Wait THEN Peak(I)=Peak(I)+Mv/2
15539  NEXT K
15542  IF Peak(I)(=.0001 THEN Peak(I)=.0001   ! Protect against apparent zero-peaks
15545  RETURN
15548 Escape_rough:Scan_all=0
15551  Escape=1
15554  MAT Peak= (0)
15557 SUBEND ! ----------------------------------------------------------------
15560 !
15563  !
15566 Interfere:SUB Interfere(U,In,Bkrds(*),INTEGER Magpos0(*),Number_peaks,Subflag,Bmc_out,Next_run)
15569  OPTION BASE 1
15572  COM /Interfere/ Interfere(*),Interf_nsecs,Interf_mon_cts(*),Interf_mon_time(*),INTEGER Long(*)
15575  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
15578  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
15581  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
15584  ! monitor for isobaric interferences
15587  DIM Sums(5)
15590  FOR I=0 TO 19
15593    ON KEY I LABEL "" CALL Clunk
15596  NEXT I
15599  IF Auto*Full_auto THEN ON KEY 0 LABEL "   BMC" GOTO Bmc_out
15602  IF Auto*Full_auto THEN ON KEY 1 LABEL "  NEXT RUN" GOTO Next_run
15605  ON KEY 9 LABEL "    ESCAPE" GOTO 15698
```

```
15251  A=POS(Spikedrun_ratio$(3),"/")
15254  IF Spikedrun_iso(3) THEN Spikedrun_ratio$(3)[A,A]="*"
15257  FOR Drive=1 TO 0 STEP -1
15260    ON ERROR GOTO 15275
15263    ASSIGN @Path1 TO "SPIKE:INTERNAL,4,"&VAL$(Drive)
15266    OUTPUT @Path1,Spike;Spike$,Spkdrun_ref,Spikedrun_iso(*),Spikedrun_ratio$(*),Iso_dif(*),Spike_ratio(*),Natural_ratio(*)
15269    ASSIGN @Path1 TO *
15272    SUBEXIT
15275  NEXT Drive
15278  DISP FNH$("UNABLE TO STORE SPIKE-DATA ON EITHER DISK")
15281  Clunk
15284  WAIT 3
15287 SUBEND ! ----------------------------------------------------------
15290 !
15293  !
15296 Rank:SUB Rank(Rp,REAL Peak(*),INTEGER L,N,Mip,Ref,Order(*),Magnet(*))
15299! sort the Peak array by descending peak-intensity & store in Order(*)
15302  FOR I=1 TO N-1
15305    X=Peak(I)
15308    FOR J=I+1 TO N
15311      IF Peak(J)<=X THEN Nextj
15314      X=Peak(J)
15317      Peak(J)=Peak(I)
15320      Peak(I)=X
15323      E=Order(I)
15326      Order(I)=Order(J)
15329      Order(J)=E
15332 Nextj:NEXT J
15335  NEXT I
15338  FOR I=1 TO N
15341    IF Magnet(Order(I),1)=Ref THEN Rp=I !Rp is order of ref-pk. in Magnet-list
15344  NEXT I
15347  Mip=Order(1) !  Mip is order of Most Intense Peak in Magnet-list
15350  L=Mip
15353  OUTPUT 0 USING "4A,4Z";"$OMU",Magnet(L,1)
15356 SUBEND ! ----------------------------------------------------------
15359 !
15362 !
15365 Rough:SUB Rough(Scan_all,Peak(*),Escape,INTEGER Data_daly,N,Order(*))
15368  !
15371  ! rapid peak-scan to determine rough peak-heights & ratios
15374  !
15377  OPTION BASE 1
15380  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
15383  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
15386  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
15389  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
15392  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
15395  INTEGER Pr
15398  FOR I=Auto*Full_auto*2 TO 19
15401    ON KEY I LABEL "" CALL Clunk
15404  NEXT I
15407  ON KEY 9 LABEL "    ESCAPE" GOTO Escape_rough
15410  OFF KBD
15413  M=Mu
15416  Subflag=0
15419  Escape=0
15422  Daly_count=0
15425  FOR I=1 TO N
15428    IF Daly_ok(Order(I)) THEN Daly_count=1+Daly_count
```

```
15608  INTEGER Pr
15611  Subflag=1
15614  FOR M=1 TO 4
15617    IF (Interfere(M,1)=0) OR Long(M) OR Interf_mon_cts(M,U/2+1) THEN 15677
15620    OUTPUT 8 USING "2(4A,4Z)";"$OFJ",MagposO(M+Number_peaks),"$OMV",Interfere(M,1)
15623    MAT Sums= (0)
15626    Zero=(Bkrds(1+U,M+Number_peaks)+Bkrds(2+U,M+Number_peaks))/2
15629    FOR K=-7 TO Interf_nsecs
15632      Enter_beam(Counts,Mv,1,1,Subflag)
15635      OUTPUT 8;"$ICL"
15638      ENTER 8;I
15641      IF Subflag>1 THEN SUBEXIT
15644      IF K>0 THEN CALL Sums(Sums(*),1,Counts,I-In,0)
15647      DISP "INTERFERENCE-MONITOR PK:     ";CHR$(129);Interfere(M,1);CHR$(128);TAB(45);CHR$(128+(K>0));Interf_nsecs-K+1;CHR$(128
);"  ";DROUND(Mv,3);"mV"
15650    NEXT K
15653    Interf_mon_cts(M,U/2+1)=Sums(1)/Interf_nsecs-Zero ! monitor peak-height in cts/sec
15656    Interf_mon_time(M,U/2+1)=Sums(3)/Interf_nsecs ! average time of mon-pk
15659    FOR K=M+1 TO 4 ! share monitor peak data for multiple-interferences
15662      IF Interfere(K,1)=Interfere(M,1) THEN
15665        Interf_mon_cts(K,U/2+1)=Interf_mon_cts(M,U/2+1)
15668        Interf_mon_time(K,U/2+1)=Interf_mon_time(M,U/2+1)
15671      END IF
15674    NEXT K
15677  NEXT M
15680  SUBEXIT
15683 Bmc_out:Bmc_out=1
15686  SUBEXIT
15689 Next_run:Next_run=1
15692  SUBEXIT
15695  !
15698  Subflag=4*(Auto=0)
15701 SUBEND! ------------------------------------------------------------
15704 !
15707 !
15710 Writedata:SUB Writedata(Aver(*),Ratio$(*),Ffile,Na$,Sigma$(*),Delta$(*),Acc(*),Pk,Time,Filcurr,INTEGER Block,Run,Sample,N,Data
_collector,Prtr(*))
15713   ! store data for block on disk
15716  PRINTER IS Prtr(2)
15719  Date$=DATE$(TIMEDATE)
15722  ON ERROR GOTO 15785
15725  OFF KBD
15728  SUSPEND INTERACTIVE  ! Lockout keyboard during data-write operations
15731  ASSIGN @Path1 TO "LRES:INTERNAL,4,1"
15734  ENTER @Path1;Lres   ! last file used
15737  Lres=Lres+1-500*(Lres>499)  ! Increment last file used (max# files is 500)
15740  ASSIGN @Path1 TO "LRES:INTERNAL,4,1"
15743  OUTPUT @Path1;Lres
15746  IF Block<=1 THEN Ffile=Lres
15749  ON ERROR GOTO 15785
15752  ASSIGN @Path1 TO "RESDIR:INTERNAL,4,1"
15755  OUTPUT @Path1,Run;Sample,Date$,Na$,Ffile,Lres
15758  ON ERROR GOTO 15794
15761  ASSIGN @Path1 TO "RESULT:INTERNAL,4,1"
15764  OUTPUT @Path1,Lres;Na$[1,10],Ratio$(*),Aver(*),Sigma$(*),Acc(*),Delta$(*),Pk,N,Data_collector,Block,Time,Filcurr
15767  ASSIGN @Path1 TO *
15770  RESUME INTERACTIVE
15773  OFF ERROR
15776  PRINTER IS CRT
15779  SUBEXIT
```

```
15782  !
15785  PRINT USING "2/,10X,5(K),2/";"**** DATA-DIRECTORY WRITE-ERROR, RUN# ",Run,"   BLOCK# ",Block," ****"
15788  GOTO 15770
15791  !
15794  PRINT USING "2/,10X,5(K),2/";"**** UNABLE TO STORE DATA FOR RUN# ",Run," , BLOCK# ",Block," ****"
15797  GOTO 15770
15800 SUBEND ! ----------------------------------------------------------------
15803 !
15806  !
15809 Printres:SUB Printres(INTEGER Printer,Run,REAL Escape,OPTIONAL REAL File1,File2)
15812   ! print out the isotope-ratio results for a run (get from disk)
15815  OPTION BASE 1
15818  INTEGER Sample,N,Data_collector,Block
15821  REAL Pk,Acc(7),Aver(7)
15824  DIM Ratio$(7)[7],Date$[12],Na$[50],D$[2],Delta$(7)[10],Sigma$(7)[10],Name_frag$[10]
15827 !OUTPUT KBD;CHR$(255)&CHR$(75);
15830  D$=" D"
15833  OFF KNOB
15836  ON KEY 9 LABEL "    ESCAPE" GOTO 15977
15839  PRINTER IS Printer
15842  IF NPAR<4 THEN
15845    ON ERROR GOTO 15953
15848    ASSIGN @Path1 TO "RESDIR:INTERNAL,4,1"
15851    ENTER @Path1,Run;Sample,Date$,Na$,Firstfile,Lastfile
15854    IF Firstfile=0 THEN Firstfile=Lastfile-10
15857    IF Printer>1 THEN PRINT USING "2(/,K),/";RPT$("X*",40),RPT$("*X",40)
15860    PRINT USING "25A,2D,7X,4A,2D,15X,K,/,K";"DATA SUMMARY FOR BARREL# ",Sample,"RUN#",Run,Date$,Na$
15863    IF Printer>1 THEN PRINT
15866  ELSE
15869    Firstfile=File1
15872    Lastfile=File2
15875  END IF
15878  PRINT "BLOCK";TAB(8);"RATIO";TAB(16);"AVERAGE";TAB(26);"SIGMA%";TAB(34);"SIGMAMEAN%";TAB(46);"DELTA%";TAB(54);"mV REF-PK";TAB
(65);"FilCurr";
15881  PRINT USING "2K,K";"Minutes"
15884  IF Printer>1 THEN PRINT
15887  FOR K=Firstfile TO Lastfile+500*(Firstfile>Lastfile)
15890    I=K-500*(K>500)
15893    ON ERROR GOTO 15962
15896    ASSIGN @Path1 TO "RESULT:INTERNAL,4,1"
15899    ENTER @Path1,I;Name_frag$,Ratio$(*),Aver(*),Sigma$(*),Acc(*),Delta$(*),Pk,N,Data_collector,Block,Time,Filcurr
15902    IF Block=0 THEN
15905      PRINT USING "2/,26X,K,2/";"***** RUN ABORTED *****"
15908    ELSE
15911      FOR J=1 TO N-1
15914        IF J=1 THEN PRINT TAB(2);VAL$(Block)&D$[1,1+Data_collector];
15917        PRINT TAB(7);Ratio$(J);TAB(15);DROUND(Aver(J),6);TAB(26);Sigma$(J);TAB(34);DROUND(Acc(J),3);TAB(46);Delta$(J);
15920        IF J=1 THEN PRINT TAB(55);DROUND(Pk,3);TAB(64);Filcurr;TAB(73);DROUND(Time,4);
15923        PRINT
15926      NEXT J
15929      IF Printer>1 THEN PRINT
15932    END IF
15935  NEXT K
15938  IF Printer>1 THEN PRINT USING "/,K,/,K,5/";RPT$("X*",40),RPT$("*X",40)
15941  IF Printer=1 THEN PRINT
15944  PRINTER IS CRT
15947  OFF ERROR
15950  SUBEXIT
15953  PRINTER IS CRT
15956  PRINT USING "/,3(K),/";"***** UNABLE TO READ DIRECTORY FOR RUN# ",Run," ******"
```

```
15959  GOTO 15947
15962  PRINTER IS CRT
15965  PRINT USING "/,7(K),/";"***** UNABLE TO READ DATA FOR BLOCK# ",I+1-Firstfile,"     RUN# ",Run,"     FILE# ",I,"  *****"
15968  PRINTER IS Printer
15971  GOTO 15935
15974  !
15977  Escape=1
15980 SUBEND ! ------------------------------------------------------------
15983 !
15986 !
15989 Timeconst_corr:SUB Timeconst_corr(Corr_(*),REAL Resdecay(*),Raw_peaks(*),INTEGER N,Integr_time(*),Delay_time(*))
15992 ! do corrections for resistor time-constants, to depth of 3 sets
15995  OPTION BASE 1
15998  REAL Avpk(8),Peak(300),Itime(8)
16001  MAT Avpk= RSUM(Raw_peaks)     ! average peak-heights for each isotope
16004  MAT Itime= Integr_time+Delay_time  ! total peak-time for each isotope
16007  Total=SUM(Itime)             ! set time
16010  T=0
16013  FOR I=1 TO N    ! array of peak-heights for each second, 4 sets deep
16016    FOR J=1 TO Itime(I)
16019      T=T+1
16022      P=Avpk(I)
16025      Peak(T)=P
16028      Peak(T+1*Total)=P
16031      Peak(T+2*Total)=P
16034      IF T+3*Total<301 THEN Peak(T+3*Total)=P
16037    NEXT J
16040  NEXT I
16043  Ct=0
16046  FOR I=1 TO N
16049    Reach=Ct+1      ! First donor-peak (1st sec. of Ith-pk 2-sets-previous)
16052    Timeconst_corr=0
16055    FOR J=1 TO Itime(I)
16058      Ct=Ct+1
16061      IF J>Delay_time(I) THEN
16064        Last=Ct+3*Total-1  ! Last donor-peak  (previous second)
16067        FOR T=Last TO Reach STEP -1
16070          Z=Last-T+1
16073          IF (Resdecay(Z)=0) OR (Z=80) THEN 16085
16076          Timeconst_corr=Timeconst_corr+Peak(T)*Resdecay(Z) ! effect of donor peak
16079        NEXT T
16082      END IF
16085    NEXT J
16088    Corr_(I)=-Timeconst_corr/Avpk(I)/Integr_time(I)    ! average fractional correction required for receiving peak
16091  NEXT I
16094  DISP
16097 SUBEND ! ------------------------------------------------------------
16100 !
16103 !
16106 Resdecay:SUB Resdecay(REAL Resistor_const(*),Resdecay(*))    ! calculate 80 seconds of resistor-decay effects
16109  FOR I=2 TO 6 STEP 2
16112    IF Resistor_const(I)=0 THEN Resistor_const(I)=1
16115  NEXT I
16118  FOR I=1 TO 80
16121    T=I-.5
16124    Resdecay(I)=Resistor_const(1)*EXP(-T/Resistor_const(2))+Resistor_const(3)*EXP(-T/Resistor_const(4))+Resistor_const(5)*EXP(-
T/Resistor_const(6))
16127    IF Resdecay(I)<1.E-8 THEN SUBEXIT  ! ignore DZC tails of <.01 ppm
16130  NEXT I
16133 SUBEND ! ------------------------------------------------------------
```

```
16136 !
16139 !
16142 Magprt:SUB Magprt(INTEGER Coarse,Magnet(*),Niso)
16145  PRINT USING "K,2D,8X,K,4D";"COARSE-MAGNET RANGE:  ",Coarse,"Re-187 Magnet-setting: ",Magnet(0,2)
16148  FOR J=0 TO 12 STEP 12
16151    FOR I=1+J TO 12+J
16154      IF I>Niso THEN 16163
16157      PRINT TAB(6*(I-J)-2);Magnet(I,1);
16160    NEXT I
16163    PRINT
16166    FOR I=1+J TO 12+J
16169      IF I>Niso THEN 16184
16172      PRINT TAB(6*(I-J)-2);Magnet(I,2);
16175    NEXT I
16178    PRINT
16181  NEXT J
16184  PRINT
16187 SUBEND ! ------------------------------------------------------------
16190 !
16193 !
16196 Decay:SUB Decay(Reduced_current,Flashed,Minbeam,Maxgrowth,Tabort,INTEGER Block)
16199 !
16202 ! monitor decay/growth of beam, flash if noisy.
16205  OPTION BASE 1
16208  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
16211  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
16214  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
16217  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
16220  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,F8
16223  DIM Sums(5)
16226 !
16229  INTEGER Pr
16232  FOR I=Auto*Full_auto*2 TO 19
16235    ON KEY I LABEL "" CALL Clunk
16238  NEXT I
16241  ON KEY 9 LABEL "    ESCAPE" GOTO 16406
16244  OFF KBD
16247  OUTPUT 8;Mm$(Mu,1)
16250  Nsec=10+5*(Block=0)
16253  Subflag=0
16256  Flashed=0
16259  IF Block=0 THEN Reduced_current=0
16262  T0=TIMEDATE
16265  MAT Sums= (0)
16268  FOR Y=0 TO Nsec
16271    Enter_beam(X,Mv,1,1,Pr)
16274    IF Mv>2 AND Y AND Mv<Last_mv/2 THEN Mv=Last_mv ! crude dropout protection
16277    Last_mv=Mv
16280    DISP "CHECKING BEAM NOISE AND DECAY    ";Nsec-Y,DROUND(Mv,4);" mV"
16283    IF Pr>1 THEN Decay
16286    IF Y THEN
16289      CALL Sums(Sums(*),1,Mv,Y,10)
16292    END IF
16295  NEXT Y
16298  CALL Slope(Sums(*),Slope,Int,Scatter_mv,Slope_sig,(Nsec))
16301  Av_beam=Sums(1)/Nsec  ! average beam in millivolts
16304  Decay=Slope*60*100/Av_beam  ! beamchange in %/minute
16307  Decay_sigma=Slope_sig*60*100/Av_beam ! uncertainty in %beam decay
16310  Noise_percent=100*Scatter_mv/Av_beam
16313  PRINT USING "/,3(K),8X,3(K),4A,#";"BEAM-CHANGE= ",DROUND(Decay,2)," %/MIN.","NOISE= ";DROUND(Noise_percent,2)," %/SEC."
```

```
16316  Theor_noise=SQR(Noise(Mu)^2+Av_beam/Ions)
16319  Noisy=(Noise_percent).5) AND (Scatter_mv>2.5*Theor_noise)  ! beam is defined as noisy if > 0.5%/second noise and > theoretica
l noise
16322  IF NOT Noisy AND (Decay+2*Decay_sig)>-10 THEN Check_growth
16325  !
16328  ! beam either noisy or decaying rapidly
16331  T=(TIMEDATE-T0)/60
16334  PRINT USING "3(K),/";"(decay or noise exceeds tolerance:   ",DROUND(Tabort-T,2)," minutes to abort or flash)"
16337  IF T<Tabort THEN 16265
16340  IF Noisy AND NOT Flashed THEN Flash
16343  Error_message(Prtr(*),3,"CAN'T PROCEED WITH RUN# "&VAL$(Run)&" --  UNSTABLE OR RAPIDLY DECAYING")
16346  Subflag=1
16349  GOTO 16409
16352   !
16355 Flash: !
16358  Error_message(Prtr(*),3,FNB1$(" NOISY BEAM ")&"     FLASHING TO "&VAL$(INT(1150*Filament(Nfils))/1000)&" AMPS")
16361  Flashed=1
16364  CALL Filament(50,1.15*Filament(Nfils),1,Nfils)      ! increase sample-filament current by 15%
16367  Wait(TIMEDATE,120,Filament(Nfils),Magnet(L,1),Auto,Full_auto) ! wait 2 minutes after flashing
16370  CALL Filament(10,Filament(Nfils)/1.15,1,Nfils) ! restore original sample-filament current
16373  Wait(TIMEDATE,180,Filament(Nfils),Magnet(L,1),Auto,Full_auto) ! wait 3 minutes
16376  GOTO 16262
16379       !
16382 Check_growth: ! is rate of beam-growth too large?
16385  IF Block=0 AND Reduced_current=0 AND (Av_beam<Minbeam)*(Growth)2*Maxgrowth) OR (Av_beam)=Minbeam)*(Growth)Maxgrowth) THEN
16388  ! reduce fil-curr if growth)MAX & beam)MIN, OR IF growth)2*MAX & beam<MIN         (before 1st data-block only)
16391    Reduce(Reduced_current,Maxgrowth,Filament(*),Nfils)    ! beam-growth too large
16394    Wait(TIMEDATE,20,Filament(Nfils),Magnet(L,1),Auto,Full_auto)
16397    ON KEY 9 LABEL "    ESCAPE" GOTO 16406
16400    GOTO 16262
16403  END IF
16406  DISP
16409  OUTPUT KBD;CHR$(255)&CHR$(75);
16412 SUBEND ! ------------------------------------------------------------
16415 !
16418  !
16421 Reduce:SUB Reduce(Reduced_current,Maxgrowth,Filament(*),Incr_block,INTEGER Nfils)
16424! reduce filament-current by 2.4% if beam-growth is too great
16427  PRINT USING "/,3(K),2/";"BEAM-GROWTH EXCEEDS ASSIGNED LIMIT OF ",Maxgrowth," %/MINUTE"
16430  CALL Filament(10,Filament(Nfils)/1.024,1,Nfils)  ! reduce fil-current by 2.4%
16433  Reduced_current=Reduced_current+1
16436  Incr_block=0
16439 SUBEND ! ------------------------------------------------------------
16442 !
16445  !
16448 Peaks:SUB Peaks(R,Peak_t(*),Set,In,Peak_height(*),Pmax,Aver(*),INTEGER N,Coarsemag,Mag_pos(*),Integr_time(*),Wait_time(*),Isot
ope(*),Dropouts)
16451  OPTION BASE 1
16454  ! This is the actual peaktop-jumping routine for data-taking, & includes real-time graphics & ratio-calculation
16457  !
16460  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
16463  COM /Keyboard/ Cn$,C1$,Cb$,Cu$,Q$,Clear$
16466  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
16469  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
16472  COM /Filaments/ Filament(4),Fil$(4)[10],INTEGER Nfils,F8
16475  COM /Data1/ Peak(*),Pk,Interfere(*),Normal(2),T00,INTEGER Data_iso(*),Data_collector,Inv,Bmc_out,Next_run,Spike
16478  COM /Data3/ Bkrds(*),Bkrd_var(*),Resdecay(*),INTEGER Sample,N_prime,Nsets,Ref,Rf,Bkrd_rdgs(*),Bkrd_sigma(*)
16481  INTEGER Pr,Int_time,Nrat(7),Nratios
16484  DIM Drat(7,38),Sums(5),Rsums(5),Tsums(5),Asums(7,5),Last_set(2,8),Decay$[22]
16487  Decay$="beam-"
```

```
16490  FOR I=0 TO 19
16493    ON KEY I LABEL "" CALL Clunk
16496  NEXT I
16499  IF Auto*Full_auto THEN ON KEY 0 LABEL "     BMC" GOTO Bmc_out
16502  IF Auto*Full_auto THEN ON KEY 1 LABEL "   NEXT RUN" GOTO Next_run
16505  ON KEY 9 LABEL "    ESCAPE" GOTO Operator_exit
16508  OFF KBD
16511  MAT Nrat= (0)
16514  MAT Rsums= (0)
16517  Dropouts=0 ! # of GPIO-interface dropouts caught
16520  Dot_interval=9
16523  PRINTER IS CRT
16526  Spac=2*(N=8)-4*(N<5)
16529  MAT Peak_height= (0)
16532  MOVE 0,0
16535  OUTPUT 8 USING "4A,4Z";"$OFJ",Mag_pos(1)
16538  WAIT 1 ! allow time for magnet to switch from far-away bkrd positions
16541  Normal_delta=0
16544  J=0
16547  MAT Asums= (0)
16550  Ref_bkrd=(Bkrds(1,R)+Bkrds(2,R))/2  ! reference-peak background
16553  FOR I=1 TO N
16556    IF I<>R THEN
16559      Ratio$=VAL$(Ref)&"/"&VAL$(Isotope(I))
16562      IF Inv=-1 THEN Ratio$=FNInvert$(Ratio$)
16565      PRINT Cu$;TABXY((13-Spac)*(I-(I)R))-(10-Spac),17);Ratio$;Cn$
16568    END IF
16571  NEXT I
16574  FOR Set=1 TO Nsets
16577    IF Set>4 THEN ON KEY 4 LABEL "   CUT SHORT" GOTO Cutshort
16580    FOR I=1 TO N
16583      MAT Sums= (0)
16586      FOR Iz=1 TO 5
16589        Rsums(Iz)=Rsums(I-(I)R),Iz)
16592      NEXT Iz
16595      OUTPUT 8 USING "2(4A,4Z)";"$OFJ",Mag_pos(I),"$OMV",Isotope(I) ! switch magnet to peak-top
16598      !
16601      FOR J=1 TO Integr_time(I)+Wait_time(I)
16604        Enter_beam(Counts,Mv,1,1,Subflag,0,Dropouts)
16607        IF Subflag>1 THEN SUBEXIT
16610        OUTPUT 8;"$ICL"
16613        ENTER 8;Tt
16616        IF Last_time AND Tt<Last_time THEN  ! Guard against GPIO dropouts - don't accept any times less than previous time
16619          FOR Ct=1 TO 3
16622            BEEP 1500+200*Ct,.04
16625          NEXT Ct
16628          Bad_time=1+Bad_time
16631          Dropouts=1+Dropouts
16634          IF Bad_time<3 THEN 16604
16637        END IF
16640        Last_time=Tt
16643        Tt=Tt-Tn
16646        Bad_time=0
16649        Otherpk_bkrd=INT((Bkrds(1,I)+Bkrds(2,I))/2) ! non-ref pk backgrd
16652        L=Wait_time(I)+1+(J>Wait_time(I))*Integr_time(I)-J
16655        Mv_peak=(Counts-Otherpk_bkrd)/Mx(Mu) ! peak-height in millivolts
16658        IF J>Wait_time(I) AND Mv_peak<Last_mv/2 AND Last_mv>(.8-.6*Mu) THEN ! Guard against GPIO dropouts, trigger being drop o
f 2x in beamsize
16661          BEEP 1500,.05
16664          BEEP 1300,.05
```

```
16667         Bad_mv=1+Bad_mv
16670         Dropouts=1+Dropouts
16673         IF Bad_mv<3 THEN 16604
16676       END IF
16679       Last_mv=Mv_peak
16682       Bad_mv=0
16685       Time=Tt/6.E+3
16688       IF J=Integr_time(I)+Wait_time(I) THEN  ! last count on peak
16691         Last_set(1,I)=Time
16694         Last_set(2,I)=Mv_peak
16697       END IF
16700       IF J>2 THEN DRAW Time,Mv_peak
16703       IF J=2 THEN
16706         IF Set>1 THEN        ! connect to last pk with dotted line
16709           MOVE Last_set(1,I),Last_set(2,I)
16712           LINE TYPE 4,Dot_interval
16715           DRAW Time,Mv_peak
16718           LINE TYPE 1
16721         ELSE
16724           MOVE Time,Mv_peak
16727         END IF
16730       END IF
16733       DISP Ci$;Isotope(I);Cn$;TAB(15);"SET#";TAB(19);Set;TAB(27);DROUND(Mv_peak,6-3*Mu);"mV";TAB(45);CHR$(128+(J>Wait_time(I)
));L;Cn$
16736       IF J>Wait_time(I) THEN CALL Sums(Sums(*),1,(Counts),Tt,0)
16739     NEXT J
16742     BEEP 500,.05
16745     Int_time=Integr_time(I)
16748     Peak_t(I,Set)=Sums(3)/Int_time ! time of Ith peak for Kth block
16751     IF Int_time<3 THEN   ! use average pk-height if <3 seconds on peak
16754       Peak_height(I,Set)=Sums(1)/Int_time
16757     ELSE
16760       CALL Slope(Sums(*),Slope,Inter,Scatter_cts,Slope_sig,Int_time)
16763        ! linear regression of peaktop counts with time
16766       Peak_height(I,Set)=Slope*Peak_t(I,Set)+Inter ! use mid-value of linear regression with time for peak-height if >=3 seco
nds on peak
16769     END IF
16772     IF Peak_height(I,Set)-Otherpk_bkrd*(I<>R)-Ref_bkrd*(I=R)<0 THEN CALL Correct(Ff$(*),Mm$(*),Filament(*),Mag_pos(I),Mu,1,Co
arsemag,Foc(*))
16775    ! restore focus if a negative peak-height (to correct for possible arc)
16778     IF (Set>1) AND (I<>R) THEN ! calculate real-time ratio
16781       L=Set+(R<I)-1          ! Lth pair of ratio-isotopes
16784       Rf1=Peak_height(R,L)-Ref_bkrd ! net cps for reference-peak
16787       Temp1=(Peak_t(R,L)-Peak_t(I,Set-1))
16790       Temp2=(Peak_height(I,Set)-Peak_height(I,Set-1))/(Peak_t(I,Set)-Peak_t(I,Set-1))
16793       Temp3=Peak_height(I,Set-1)+Temp1*Temp2-Otherpk_bkrd ! interpolated non-ref. peak net-cps
16796       IF Temp3=0 THEN Temp3=1.E-12! Guard against div. by 0
16799       Rr=Rf1/Temp3 ! raw ratio (uncorrected for fractionation)
16802       IF NOT Spike AND Normal(1) THEN Rr=Rr/(1+(Isotope(I)<>Normal(1))*Normal_delta*(Ref-Isotope(I))/(Ref-Normal(1)))! real-t
ime ratio
16805    ! correct for mass-discrimination (linear)
16808       IF Normal_delta OR Normal(1)=0 OR Spike OR Isotope(I)=Normal(1) THEN
16811        ! calculate & display real-time ratios
16814         Ix=I-(I>R)! ratio#
16817         Nrat(Ix)=1+Nrat(Ix) !    #ratios taken for Ixth ratio
16820         Drat(Ix,Nrat(Ix))=Rr^Inv! ratio
16823         CALL Sums(Rsums(*),1,0,Drat(Ix,Nrat(Ix)),Nrat(Ix))
16826         Aver(Ix)=Rsums(3)/Nrat(Ix)! no-outliers average
16829         IF Nrat(Ix)>1 THEN
16832           Calc_sigma(Sigma,Rsums(3),Rsums(4),Nrat(Ix))
```

```
16835           IF Nrat(Ix)>4 THEN ! reject outliers at 1.7-sigma level
16838             MAT Tsums= Rsums
16841             Nratios=Nrat(Ix)
16844             FOR Iz=1 TO Nrat(Ix)
16847               IF ABS(Drat(Ix,Iz)-Aver(Ix))>1.7*Sigma THEN CALL Sums(Tsums(*),-1,0,Drat(Ix,Iz),Nratios)
16850             NEXT Iz
16853             Aver(Ix)=Tsums(3)/Nratios
16856             Calc_sigma(Sigma,Tsums(3),Tsums(4),Nratios)
16859           END IF
16862         END IF
16865         IF NOT Spike AND Isotope(I)=Normal(1) THEN Normal_delta=Aver(Ix)^Inv/Normal(2)-1   ! difference w. normalizing ratio
16868         PRINT TABXY(Ix*(13-Spac)-(10-Spac)-2*(N<5),18);VAL$(DROUND(Aver(Ix),5));
16871         IF N<5 THEN ! include std-deviations in realtime ratio-printout
16874           IF Nrat(Ix)>1 THEN PRINT "-"&VAL$(DROUND(100*Sigma/Aver(Ix),2))&"%   "
16877         ELSE
16880           PRINT "   "
16883         END IF
16886         FOR Iz=1 TO 5
16889           Asums(Ix,Iz)=Rsums(Iz)
16892         NEXT Iz
16895       END IF
16898     END IF
16901     IF Set>1 AND N<6 THEN ! display beam-growth/decay ( if room)
16904       Pk1=(Peak_height(I,Set)-Otherpk_bkrd)/Mx(Mu)
16907       Pk0=(Peak_height(I,Set-1)-Otherpk_bkrd)/Mx(Mu)
16910       IF Pk1>10-9*Mu THEN
16913         G=1.E+4*60*(Pk1/Pk0-1)/(Peak_t(I,Set)-Peak_t(I,Set-1))
16916         IF ABS(G)<.01 THEN G=0
16919         IF G>=0 THEN Decay$[6]="growth="
16922         IF G<0 THEN Decay$[6]="decay= "
16925         Decay$[13,22]=VAL$(ABS(DROUND(G,2)))&"%/min."
16928         PRINT TABXY(58,18);Decay$
16931       END IF
16934     END IF
16937   NEXT I
16940 NEXT Set
16943 SUBEXIT
16946 !
16949 Bmc_out:Bmc_out=1
16952 SUBEXIT
16955 Next_run:Next_run=1
16958 SUBEXIT
16961 !
16964 Cutshort: ! k4 key pressed: stop taking data & go on to bkrds
16967 Set=Set-1
16970 Nsets=Set
16973 Broop
16976 SUBEXIT
16979 !
16982 Operator_exit:Subflag=4
16985 SUBEND! ------------------------------------------------------------
16988 !
16991 !
16994 Calcav:SUB Calcav(Ratio(*),Time(*),Print,R$,Percent_error(*),Candump,INTEGER N,Prtr(*))
16997 OPTION BASE 1
17000 ! calculate weighted averages of ratios in a block
17003 DIM Inverse_var(80)
17006 Nn=N
17009 Count=0
17012 Recalc:Ext_sigma=0
```

```
17015  Count=Count+1
17018  Weight=0
17021  Sum_wtd_ratios=0
17024  Q=0
17027  FOR I=1 TO N
17030    IF Ratio(I) THEN
17033      Inverse_var(I)=(100/(Percent_error(I)*Ratio(I)))^2
17036      Weight=Weight+Inverse_var(I)
17039      Sum_wtd_ratios=Sum_wtd_ratios+Inverse_var(I)*Ratio(I)
17042      Q=Q+Inverse_var(I)*Ratio(I)^2
17045    END IF
17048  NEXT I
17051  Nu=Nn-1                    ! degrees of freedom
17054  T=FNSt(Nn)
17057  Wtd_aver_int=Sum_wtd_ratios/Weight ! "internal" error of wtd average
17060  Wtd_aver=Wtd_aver_int
17063  Sums=Q-Weight*Wtd_aver_int^2   ! sums of squares of weighted deviates
17066  IF Sums<0 THEN Sums=0
17069  Mswd=Sums/Nu ! mean square of weighted deviates
17072  Percent_int_sig=100/Wtd_aver_int/SQR(Weight)
17075  Percent_totsig=Percent_int_sig*SQR(Mswd)
17078  IF Mswd>100 THEN
17081    Probability=0
17084  ELSE
17087    !
17090    D=1 ! calculate the probability that internal errors alone wil give the observed scatter, using cumulative chi-square distr
ibution
17093    A=1
17096    IF NOT FNEven(Nu) THEN
17099      I=0 ! nu odd
17102      REPEAT
17105        I=I+2
17108        D=D*Sums/(Nu+I)
17111        A=A+D
17114      UNTIL D<1.E-6
17117      D=SQR(PI)
17120      FOR I=1/2 TO Nu/2
17123        D=D*I
17126      NEXT I
17129      Probability=1-EXP(-Sums/2)*(Sums/2)^(Nu/2)*A/D
17132    ELSE
17135    !
17138      FOR I=1 TO Nu/2-1! Nu even
17141        D=D*Sums/(2*I)
17144        A=A+D
17147      NEXT I
17150      Probability=A*EXP(-Sums/2)
17153    END IF
17156  END IF
17159    !
17162  Percent_err_95=Percent_int_sig*((Nn)=6)*T*SQR(Sums/Nu)+(Nn(6)*1.96)
17165  !
17168  IF Probability(.2 THEN   !calculate excess variance (Troutman's equation)
17171    Ext_sigma=SQR((Percent_totsig^2-Percent_int_sig^2)*Nn)
17174    Wtd_aver_ext=Wtd_aver_int! wtd average taking into account external variance
17177    G=0
17180    N0=0
17183    S1=0
17186  !
17189    LOOP
```

```
17192     ON ERROR GOTO No_convergence
17195     NO=1+NO
17198     DATA 0,0,0,0,0,0,0
17201     RESTORE 17198
17204     READ M,K,Q,R,D,E,L
17207     FOR I=1 TO N
17210       IF Ratio(I) THEN
17213         R=G+1/Inverse_var(I)
17216         L=(Ratio(I)-Wtd_aver_ext)^2
17219         K=K+Ratio(I)/R
17222         M=M+1/R
17225         Q=Q+L/R^2
17228         D=D+1/R^2
17231         E=E+L/R^3
17234       END IF
17237     NEXT I
17240     S1=G-(Q-M)/(D-2*E)
17243     Wtd_aver_ext=K/M
17246     DISP "iteration";NO;"    mean";DROUND(Wtd_aver_ext,6);"    ext. var.";DROUND(S1,3)
17249   EXIT IF (ABS((S1-G)/S1)<1.E-4) AND (S1>0)
17252     IF NO>30 THEN ! if hasn't converged within 30 iterations, test for convergence
17255       IF (ABS(S1-G)>=Z0) OR (S1<0) THEN
17258 No_convergence:OFF ERROR
17261         Percent_err_95=SQR((1.96*Percent_int_sig)^2+T^2*(Percent_totsig^2-Percent_int_sig^2))
17264         GOTO Reject
17267       END IF
17270     END IF
17273     Z0=ABS(S1-G)
17276     G=S1
17279   END LOOP
17282   !
17285   Ext_sigma=100*SQR(S1*Nn/Nu)/Wtd_aver_int
17288   Percent_err_95=SQR((1.96*Percent_int_sig)^2+(T*Ext_sigma)^2/Nn)
17291   Wtd_aver=Wtd_aver_ext
17294  !
17297 Reject:NO=Nn   ! reject outliers
17300   FOR I=1 TO N
17303     IF Ratio(I) AND Nn>.7*N THEN ! don't reject more than 30% of ratios
17306       Tolerance=(1+(Count-1)/8)*SQR((.02*Ratio(I)*Percent_error(I))^2+(.01*Wtd_aver*T*Ext_sigma)^2)! start rej. at 2-sigma, i
ncrease slightly each pass
17309       IF ABS(Ratio(I)-Wtd_aver)>Tolerance THEN
17312         IF Rej_print=0 THEN
17315           PRINT "REJECTED:  ";
17318           Rej_print=1
17321         END IF
17324         PRINT DROUND(Ratio(I),6);
17327         AREA COLOR 0,0,0! make the data-point box into an empty one
17330         Boxwidth=2.5
17333         Boxheight=(4*Percent_error(I)*Ratio(I)/100)
17336         MOVE Time(I)-Boxwidth/2,Ratio(I)-Boxheight/2
17339         RECTANGLE Boxwidth,Boxheight,FILL,EDGE
17342         AREA COLOR 1,1,1
17345         RECTANGLE Boxwidth,Boxheight
17348         Ratio(I)=0
17351         Nn=Nn-1
17354       END IF
17357     END IF
17360   NEXT I
17363   IF Nn<NO THEN ! recalculate wtd average
17366     DISP
```

```
17369      PRINT
17372      GOTO Recalc
17375    END IF
17378  END IF
17381    !
17384  Abs_err_95=DROUND(.01*Percent_err_95*Wtd_aver,2)
17387  IF (Abs_err_95<=0) OR (Wtd_aver<=0) THEN D=6
17390  IF (Abs_err_95>0) AND (Wtd_aver>0) THEN D=1-INT(LGT(10^INT(LGT(Abs_err_95+(Abs_err_95=0)))/Wtd_aver))
17393  FOR Y=0 TO Print
17396    PRINTER IS Prtr(Y+1)
17399    IF Y=0 THEN PRINT TABXY(1,15);
17402    IF Y THEN PRINT RPT$("-",80)
17405    PRINT USING "3(K),#";" WTD AVERAGE "&R$&" =",DROUND(Wtd_aver,D)," +/- "
17408    PRINT USING "K,2X,A,K,K,10X,K";Abs_err_95,"(",DROUND(Percent_err_95,2),"%)","(95% CONF. LIMIT) "
17411    IF Y THEN PRINT RPT$("-",80)
17414    IF Nu=0 THEN SUBEXIT
17417    PRINT USING "3(K),10X,3(K)";"INTERNAL SIGMAMEAN = ",DROUND(Percent_int_sig,2),"%","EST. TOTAL SIGMAMEAN =",DROUND(Percent_t
otsig,2),"%"
17420    IF Ext_sigma THEN PRINT "EXTERNAL SIGMA =";DROUND(Ext_sigma,2);"%"
17423    PRINT "M.S.W.D. =";DROUND(Mswd,3);TAB(37);"PROBABILITY =";DROUND(Probability*(Probability).001),2)
17426    IF Y THEN PRINT USING "80A,2/";RPT$("*",80)
17429  NEXT Y
17432  PRINTER IS CRT
17435  Tspred=Time(N)-Time(1)
17438  MOVE Time(1)-1.2-Tspred/10,Wtd_aver
17441  DRAW Time(N)+1.2+Tspred/10,Wtd_aver
17444 SUBEND ! ------------------------------------------------------------
17447 !
17450 !
17453 Peaktime:SUB Peaktime(REAL Peak(*),Noise(*),Ions,INTEGER Integr_time(*),Wait_time(*),N,Collector,Refpk,Isotope(*))
17456  ! calculate optiminum (in the sense of best precision in least time for a perfectly-stable beam) integration & wait times for
 each peak,
17459  ! using the algorithms of Ludwig, USGS Prof. Paper ???? #####
17462  OPTION BASE 1
17465  REAL Optim_timeratio(8),Integr_time0(8)
17468  MAT Optim_timeratio= (0)
17471  MAT Integr_time= (0)
17474  Mv_av_pk=.001*SUM(Peak)/N  ! average peak-height in volts
17477  A=1+99*Collector
17480  Av_integr_time=3+(Mv_av_pk(4/A)+(Mv_av_pk(1/A)+(Mv_av_pk(.1/A)+(Mv_av_pk(.01/A)+(Mv_av_pk(.001/A)
17483    !
17486  FOR I=1 TO N ! theoretically optimum ratio of integration-times
17489    Optim_timeratio(I)=SQR(((Noise(Collector)/Peak(I))^2+1/Ions/Peak(I))/((Noise(Collector)/Peak(Refpk))^2+1/Ions/Peak(Refpk)))
17492  NEXT I
17495    !
17498  ! don't nest this loop within above loop!!! must have all Optim_timeratios!!
17501  FOR I=1 TO N ! theoretically optimum integration times
17504    Integr_time0(I)=Optim_timeratio(I)*Av_integr_time*N/SUM(Optim_timeratio)
17507  NEXT I
17510  IF Integr_time0(1)<2 THEN
17513    IF N<4 AND Optim_timeratio(N)>5 THEN Av_integr_time=1.5*Av_integr_time
17516  END IF
17519    !
17522  ! for >3 isotopes, Least-Intense Peak IT no longer than next-LIP
17525  ! for 3 isotopes, LIP IT no more than 2.4 times next-LIP
17528  IF N>3 THEN Integr_time0(N)=Integr_time0(N-1)
17531  IF (N=3) AND (Integr_time0(3)>2.4*Integr_time0(2)) THEN Integr_time0(3)=2.4*Integr_time0(2)
17534  ! renormalize for average integration time
17537  MAT Integr_time0= Integr_time0*(Av_integr_time*N/SUM(Integr_time0))
17540    !
```

```
17543  FOR I=1 TO N
17546    Integr_time(I)=INT(Integr_time0(I))+(FRACT(Integr_time0(I)))=.5)
17549    IF Integr_time(I)<2 THEN Integr_time(I)=2
17552    IF Integr_time(I)>14 THEN Integr_time(I)=14
17555  NEXT I
17558   !
17561  FOR I=1 TO N
17564    Peakratio=ABS(Peak(I-1+N*(I=1))/Peak(I))
17567    Mass_jump=ABS((Isotope(I-1+N*(I=1))-Isotope(I))/Isotope(I))
17570    Wait_time(I)=1+(Peakratio>2.5)+(Peakratio>20)+(Peakratio>100)+(Peakratio>500)+(Peakratio>2000)
17573    IF (Mass_jump)>.15) AND (Wait_time(I)<3) THEN Wait_time(I)=3 ! wait at least 3 sec for large mass-jumps
17576  NEXT I
17579 SUBEND ! ------------------------------------------------------------
17582 !
17585 !
17588 Hv:SUB Hv(Mm$(*),Hv,Pass,INTEGER Mu,I_t)  ! query accelerating voltage
17591  OFF KEY
17594  OFF KNOB
17597  ON KEY 9 LABEL "    ESCAPE" GOTO 17690
17600  Pass=0
17603  V$=Mm$(Mu,1)[8]
17606  PRINT "HV =";
17609   !
17612   ! read zero
17615  OUTPUT 8;"$OMW04:"&V$
17618  FOR I=1 TO 10
17621    WAIT .2
17624  NEXT I
17627  OUTPUT 8;"$IDV"
17630  ENTER 8;Zero
17633   !
17636   ! read gross value
17639  OUTPUT 8;"$OMW0=:"&V$
17642  FOR I=1 TO 10
17645    WAIT .2
17648  NEXT I
17651  OUTPUT 8;"$IDV"
17654  ENTER 8;V
17657   !
17660  Hv=INT((V-Zero)/100)
17663  OUTPUT 8;Mm$(Mu,I_t)  ! reset system monitor to ion-collector
17666  IF Hv>1.E+4 THEN      ! impossible value
17669    Clunk
17672    DISP "GAIN MUST BE SET ON x1  -PRESS CONTINUE WHEN READY"
17675    PAUSE
17678    GOTO 17606
17681  END IF
17684  PRINT Hv;"VOLTS"
17687  SUBEXIT
17690  Pass=1
17693 SUBEND ! ------------------------------------------------------------
17696 !
17699 !
17702 Spike_check:SUB Spike_check(Spike,INTEGER Iso(*),Spikedrun_iso(*),Spkdrun_ref)  ! check if the run-variables specify the isoto
pes required by the spike
17705  FOR I=1 TO 8
17708    Q=Iso(I)
17711    IF Q THEN  ! check only for nonzero isotopes
17714      IF (Q<>Spikedrun_iso(1)) AND (Q<>Spikedrun_iso(2)) AND (Q<>Spikedrun_iso(3)) AND (Q<>Spkdrun_ref) THEN Spike=0
17717      IF Q=Spkdrun_ref THEN Spkdrun_refiso=1! ref. isotope is present
```

```
17720      IF (Q=Spikedrun_iso(1)) OR (Q=Spikedrun_iso(2)) THEN Normiso=1+Normiso! # of normalizing isotopes present
17723    END IF
17726  NEXT I
17729  IF (Spkdrun_refiso=0) OR (Normiso(2) THEN Spike=0
17732 SUBEND ! ------------------------------------------------------------
17735 !
17738 !
17741 Wait:SUB Wait(Tzero,Waitsec,Filcurr,INTEGER Isotope,Auto,Full_auto)  ! wait for Waitsec seconds
17744  FOR I=2*Auto*Full_auto TO 19
17747    ON KEY I LABEL "" CALL Clunk
17750  NEXT I
17753  ON KEY 3 LABEL "  HALVE WAIT" GOTO Halve
17756  ON KEY 4 LABEL "  DOUBLE WAIT" GOTO Double
17759  ON KEY 9 LABEL "    ESCAPE" GOTO 17831
17762  ON TIME (Tzero+Waitsec) MOD 86400 GOTO 17834
17765   !
17768  Iso$="("&VAL$(Isotope)&")"
17771  IF Isotope=0 THEN Iso$=""
17774  IF Waitsec(=60 THEN
17777    T$="SECONDS"
17780    Divide=1
17783  ELSE
17786    T$="MINUTES"
17789    Divide=60
17792  END IF
17795  IMAGE "WAITING",X,4D.2D,2X,7A,8X,K," AMPS",10X,K
17798  T=Waitsec-TIMEDATE+Tzero
17801  IF T<0 THEN 17831
17804  DISP USING 17795;T/Divide,T$,Filcurr,Iso$
17807  WAIT .1
17810  GOTO 17798
17813   !
17816 Halve:Waitsec=Waitsec/2
17819  GOTO 17762
17822 Double:Waitsec=Waitsec*2
17825  GOTO 17762
17828   !
17831  Tzero=TIMEDATE
17834  DISP
17837 SUBEND ! ------------------------------------------------------------
17840 !
17843 !
17846 Ratiocalc:SUB Ratiocalc(I,IO,Calc_inverted,R,Daly_discr,Ratio_count,Pb4678_cycle,Normal(*),Ns,Decay,INTEGER Nsets,Pb_4678,Ref,
Inv,Mu,Ratio_isotope(*))
17849  OPTION BASE 1
17852  COM /Data2/ Acc(*),Lacc(*),Sigma$(*),Delta$(*),Aver(*),Last_aver(*),Last_ratio$(*),Ratio$(*)
17855  COM /Rat/ Slope,Int,Spike,Sig,Peak_t(*),Timecon_pkcorr(*),Peak_height(*),Normrat_slope,Normrat_inter,Norm_inverted
17858  DIM Iso_ratio(39),Ratio_time(39),Ri$[7],Sums(5)
17861  INTEGER Nratios_used
17864  OFF KEY
17867  OFF KBD
17870  A=KNOB%
17873  Printsets=0 ! don't printout the individual set ratios
17876  Ratio$(I)=VAL$(Ref)&"/"&VAL$(Ratio_isotope(IO))
17879  IF Inv=-1 THEN Ratio$(I)=FNInvert$(Ratio$(I))
17882  P=(R<IO)
17885  A=P*R+(NOT P)*IO
17888  B=P*IO+(NOT P)*R
17891  FOR J=1 TO Nsets-1    ! Dodson's interpolation algorithm
17894    T=(Peak_t(A,J)+Peak_t(A,J+1)+Peak_t(B,J)+Peak_t(B,J+1))/4 ! time of interpolation
```

```
17897    Pa=Peak_height(A,J)+(T-Peak_t(A,J))/(Peak_t(A,J+1)-Peak_t(A,J))*(Peak_height(A,J+1)-Peak_height(A,J)) ! interpolation peak
A
17900    Pb=Peak_height(B,J)+(T-Peak_t(B,J))/(Peak_t(B,J+1)-Peak_t(B,J))*(Peak_height(B,J+1)-Peak_height(B,J)) ! interpolation peak
B
17903    Iso_ratio(J)=(Pa/Pb)^(2*P-1)  ! ratio of reference-pk to Ith-pk
17906    Ratio_time(J)=(T-Peak_t(1,1))/100   ! time for this ratio, in seconds from the first peak
17909  NEXT J
17912  Timecon_ratcorr=(1+Timecon_pkcorr(R))/(1+Timecon_pkcorr(IO))  ! Correction for resistor time-constants
17915  MAT Iso_ratio= Iso_ratio*(Timecon_ratcorr)
17918  Nratios_used=Nsets-1
17921  Ns=Nratios_used
17924  DATA 0,0,0,0,0
17927  RESTORE 17924
17930  READ F,Avdiscr,Z,Alpha,Normalized
17933  MAT Sums= (0)
17936  IF NOT Spike AND Normal(1) AND Ratio_count>0 THEN
17939!
17942! correct for mass-dependent fractionation using the exponential law of Russell & others, Geoch. Cosmoch. Acta, v. 42, p. 1075-
1090, 1978
17945!
17948    Normalized=1
17951    D=Ref*LOG(Normal(1)/Ref)
17954    FOR J=1 TO Ns
17957      F=((Normrat_slope*Ratio_time(J)+Normrat_inter)^Norm_inverted)/Normal(2)
17960      IF F>0 THEN Alpha=LOG(F)/D
17963      Avdiscr=Avdiscr+Alpha/Ns
17966      M=(Ref/Ratio_isotope(IO))^(Alpha*Ref)
17969      Z=Z+M/Ns
17972      Iso_ratio(J)=Iso_ratio(J)*M
17975    NEXT J
17978    Ri$=Ratio$(1)
17981    IF Inv=-1 THEN Ri$=FNInvert$(Ri$)
17984  END IF
17987   !
17990  Calc_inverted=1-2*(Iso_ratio(1)>1)  ! ratios > 1?
17993  Ik=Calc_inverted*Inv
17996  PRINTER IS 701;WIDTH (132)
17999  PRINT USING "#,2(K)";CHR$(27)&"&k2S",CHR$(27)&"&l6D"  ! small print, 1/6" line-spacing
18002  FOR J=1 TO Ns  ! calculate std deviation of ratios, reject outliers
18005    IF Printsets THEN PRINT DROUND(Iso_ratio(J)^Inv,6);
18008    Iso_ratio(J)=Iso_ratio(J)^Calc_inverted  ! convert to reciprocal, raw ratios >1
18011    CALL Sums(Sums(*),1,Iso_ratio(J),Ratio_time(J),0)
18014  NEXT J
18017  Rej_tolerance=2  ! rejection tolerance, in sigma
18020  Number_passes=1  ! how many passes thru set-ratios for rejection
18023  Nrej=0           ! # rejected ratios
18026  !
18029  ! ratio-rejection routine ------------------------------------------
18032  LOOP
18035    K1=Nratios_used
18038    Calc_sigma(S1,Sums(1),Sums(2),Nratios_used)
18041    Rr=Sums(1)/Nratios_used
18044    FOR J=1 TO Ns
18047      IF (ABS(Rr-Iso_ratio(J)))>Rej_tolerance*S1) AND Iso_ratio(J) THEN
18050        Nrej=1+Nrej
18053        IF Printsets AND Nratios_used=Ns THEN
18056          PRINT
18059          PRINT "REJECTED  ";
18062        END IF
18065        IF Printsets THEN PRINT DROUND(Iso_ratio(J)^Ik,6);
```

```
18068       CALL Sums(Sums(*),-1,Iso_ratio(J),Ratio_time(J),Nratios_used)
18071       Iso_ratio(J)=0
18074     END IF
18077   NEXT J
18080   IF Number_passes>2 THEN Rej_tolerance=Rej_tolerance+.3! increase rejection-tolerance by .3-sigma with each pass after the s
econd pass
18083   Number_passes=Number_passes+1
18086 EXIT IF Nratios_used=K1 OR Nratios_used<.7*Ns OR Number_passes=10
18089 ! don't reject more than 30% of ratios
18092 END LOOP
18095 ! ------------------------------------------------------------
18098 !
18101 Sig=100*S1/Rr
18104 Aver(I)=Rr^Ik   ! correct for Daly discrimination here, if desired
18107 Ns=Nratios_used+1
18110 IF Pb_4678 AND (Pb4678_cycle()2) THEN PRINT "(partial-block 206/204 =";DROUND(Rr^Ik,6);", sigma% obs. =";DROUND(Sig,3);"%)"
18113 CALL Slope(Sums(*),Slope,Int,S0,S9,Nratios_used)
18116 PRINT USING "K,#";CHR$(27)&"&l6D"  ! 1/6" line spacing
18119 PRINT "REJECTED";Nrej;"RATIO(S) OUT OF";Nsets-1;TAB(58);"TIME-CONSTANT CORRECTION =";DROUND(1.E+6*(Timecon_ratcorr^Inv-1),3);
"PPM";
18122 IF (I=1 AND NOT Pb_4678) OR (I>1 AND Pb_4678) THEN
18125   PRINT TAB(103);"BEAM-";
18128   IF Decay>=0 THEN PRINT "GROWTH";
18131   IF Decay<0 THEN PRINT "DECAY";
18134   PRINT " = ";DROUND(ABS(Decay),2);"%/minute"
18137 ELSE
18140   PRINT
18143 END IF
18146 IF Normalized THEN
18149   PRINT "MASS-DISCRIMINATION CORRECTION WAS ";DROUND(100*Avdiscr,3);"%/A.M.U.      ";
18152   PRINT "(discr.-raw ratio was";DROUND((Rr*Z^(-Calc_inverted))^Ik,6);")"
18155 END IF
18158 IF Daly_discr THEN PRINT "AVERAGE CORRECTED FOR DALY DISCRIM. OF";Daly_discr;"%/A.M.U.    (RAW =";DROUND(Rr^Ik,6);")"
18161 IF ABS(Slope)>3*S9 THEN PRINT USING "3(K),8X";"<<<<<< RATIO CHANGE DURING BLOCK OF ",DROUND(6000*Slope*Ik/Rr,2)," % PER MINUT
E >>>>>>"
18164 PRINT USING "#,2(K)";CHR$(27)&"&k0S",CHR$(27)&"&l8D"   ! normal print
18167 IF NOT Pb_4678 OR Pb4678_cycle>1 THEN PRINT
18170 SUBEND ! ------------------------------------------------------------
18173 !
18176 !
18179 Backgrounds:SUB Backgrounds(Comzer,Number_peaks,Peak(*),R,Pb4678_cycle,Nn,After,INTEGER N,Pb_4678,Integr_time(*),Bkrd_posn(*),
Ratio_isotope(*),Coarsemag)
18182 OPTION BASE 1
18185 COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
18188 COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
18191 COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
18194 COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
18197 COM /Filaments/ Filament(4),Fil$(4)[10],INTEGER Nfils,F8
18200 COM /Data1/ Peak_in(*),Pk,Interfere(*),Normal(*),TOO,INTEGER Data_iso(*),Data_collector,Inv,Bmc_out,Next_run,Spike
18203 COM /Data3/ Bkrds(*),Bkrd_var(*),Resdecay(*),INTEGER Sample,N_prime,Nsets,Ref,Rf,Bkrd_rdgs(*),Bkrd_sigma(*)
18206 REAL Bkrd_cts(120),Sums(5)
18209 INTEGER Bkrd_integr,Bkrd_time(12)
18212 Rejprint=0 ! don't print rejected bkgrds
18215 FOR I=0 TO 19
18218   ON KEY I LABEL "" CALL Clunk
18221 NEXT I
18224 IF Auto*Full_auto THEN ON KEY 0 LABEL "     BMC" GOTO Bmc_out
18227 IF Auto*Full_auto THEN ON KEY 1 LABEL "   NEXT RUN" GOTO Next_run
18230 ON KEY 9 LABEL "    ESCAPE" GOTO 18656
18233 Correct(Ff$(*),Mm$(*),Filament(*),Bkrd_posn(1,1),Mu,1,Coarsemag,Foc(*))
```

```
18236  FOR I=1 TO Nn
18239    IF (Pb_4678=0) OR Pb_4678*((Pb4678_cycle=0)+Pb4678_cycle*(After=2)+(I<>R)) THEN
18242      Bkrds(1+After,I)=0  ! Zero bkgrd elements except for shared 206 of Pb6784 blocks
18245      Bkrds(2+After,I)=0
18248    END IF
18251  NEXT I
18254  MAT Bkrd_time= (Nsets/2)  ! calculate bkrd times, use Nsets/2 for interf-monitor peaks
18257  IF Comzer THEN
18260    A=0  ! calculate optimum background for shared positions
18263    FOR I=1 TO Number_peaks
18266      IF I<>R THEN
18269        Ratio=Integr_time(I)/Integr_time(R)
18272        C=Nsets*(Peak(R)-Peak(I))/(Peak(R)*Peak(I))*SQR(1+Ratio)
18275        D=1/Peak(R)^2+1/(Ratio*Peak(I)^2)+(1/Peak(R)+1/(Ratio*Peak(I)))/(Ions*Noise(Mu)^2)
18278        Bcom=MAX((A),C/SQR(D))
18281        A=Bcom
18284      END IF
18287    NEXT I
18290  END IF
18293  FOR I=1 TO Number_peaks
18296    IF Pb_4678 AND Pb4678_cycle<>1 AND I=2 THEN
18299      Bkrd_iso=4
18302    ELSE
18305      Bkrd_iso=I
18308    END IF
18311     ! optimum background times
18314    Bkrd_time(I)=Comzer*(Bcom*Integr_time(R)/4)+(NOT Comzer)*(Nsets*Integr_time(I)/4*SQR(Noise(Mu)^2/(Noise(Mu)^2+ABS(Peak(Bkrd
_iso))/Ions)))
18317    IF Mu AND Peak(Bkrd_iso)<.4 AND Bkrd_time(I)<7 THEN Bkrd_time(I)=7      ! at least 7 seconds for small Daly peaks
18320  NEXT I
18323  Rejected=0
18326  Bct=0
18329  FOR I=1 TO Nn    ! take the background data
18332    IF Comzer=0 OR I>=Number_peaks THEN ! skip all but the backgrounds for the least-intense peak if Comzer=1 but not incl. int
erf-mon pks
18335      FOR P=1+After TO 2+After! After=0 for before-peaktop bkrds, 2 for after
18338        Above=(P=2)+(P=4)! =1 for above bkrds, 0 for below bkrds
18341        MAT Sums= (0)
18344        Posn=DROUND(Ratio_isotope(I)+(2*Above-1)/2,4)!mass-posn of this bkrd
18347        IF (Ratio_isotope(I)=0) OR Bkrds(P,I) THEN Next_p ! skip if this position shared or if no isotope specified
18350        Bkrds(P,I)=0
18353        L=0
18356        OUTPUT 8 USING "4A,4Z";"$OFJ",Bkrd_posn(I,Above+1)
18359        Bct=1+Bct! to tell when the first mag-switch has occurred
18362        U=4+(Bct=1)*(3+(After>0))! bkrds wait-time (+3 or 4 for 1st reading)
18365      !
18368        Bkrd_integr=Bkrd_time(I)
18371        FOR Si=-1 TO 1 STEP 2! look up & down isotope for shared bkrd-posns
18374          FOR J=1 TO Nn
18377            L=J*(Ratio_isotope(I)=Ratio_isotope(J)+Si)! L is position in isotope list of sharing isotope
18380            IF L THEN ! if found a shared position
18383              ! Bkrd_integr is the #secs to be spent on background (max if shared)
18386              Bkrd_integr=MAX(Bkrd_time(I),((NOT Above)*(Si=1)+Above*(Si=-1))*(L<>0)*Bkrd_time(L+(NOT L)))
18389              GOTO 18404
18392            END IF
18395          NEXT J
18398        NEXT Si
18401      !
18404        IF Bkrd_integr>120 THEN Bkrd_integr=120 ! no more than 120 seconds on this background
18407        IF Bkrd_integr<5 THEN Bkrd_integr=5 ! no less than 5 seconds
```

```
18410       IF (Bkrd_integr(6) AND Comzer THEN Bkrd_integr=6! )=6 seconds if a shared position
18413       FOR J=1 TO U+Bkrd_integr
18416         Enter_beam(Counts,Mv,1,1,Subflag)
18419         IF Subflag>1 THEN SUBEXIT
18422         DISP Ci$;Ratio_isotope(I)+(2*Above-1)/2;Cn$;TAB(20);Counts;TAB(30);CHR$(128+(J>U));U+(J>U)*Bkrd_integr-J+1;Cn$
18425         IF J>U THEN
18428           Bkrd_cts(J-U)=Counts
18431           CALL Sums(Sums(*),1,(Counts),0,Bkrd_integr)
18434         END IF
18437       NEXT J
18440       Calc_sigma(Sigma,Sums(1),Sums(2),Bkrd_integr)
18443       FOR J=1 TO Bkrd_integr
18446      ! reject background readings at a 2-sigma tolerance
18449         IF (ABS(Sums(1)/Bkrd_integr-Bkrd_cts(J)))>2*Sigma) AND Sigma THEN
18452           IF NOT Rejected AND Rejprint THEN
18455             PRINT "REJECTED:  ";
18458             Rejected=1
18461           END IF
18464           IF Rejprint THEN PRINT VAL$(Bkrd_cts(J))&" ("&VAL$(Ratio_isotope(I)+(2*Above-1)/2)&")    ";
18467           CALL Sums(Sums(*),-1,(Bkrd_cts(J)),0,0)
18470         END IF
18473       NEXT J
18476       Bkrds(P,I)=Sums(1)/Bkrd_integr  ! average bkrd cts/sec
18479       Calc_sigma(Sigma,Sums(1),Sums(2),Bkrd_integr)
18482       Bkrd_sigma(P,I)=Sigma     ! std dev. of bkrd counts/sec
18485       Bkrd_rdgs(P,I)=Bkrd_integr ! # of readings used for std-dev-counts/sec
18488       FOR J=1 TO Mn
18491        !share bkrds with other pks that have bkrds at same mass-position
18494         FOR K=1 TO 2! search other-pk above & below posns
18497           IF DROUND(Ratio_isotope(J)+K-1.5,4)=Posn THEN GOSUB Share_bkrds
18500         NEXT K
18503       NEXT J
18506       BEEP 440,.05
18509 Next_p:NEXT P
18512     END IF
18515   NEXT I
18518   IF Comzer THEN
18521     FOR I=1 TO Number_peaks  ! share bkrds for least-intense peak for Comzer=1 (non-interf-mon-pks only)
18524       FOR J=1 TO 2
18527         Bkrds(J+After,I)=Bkrds(J+After,Number_peaks)
18530         Bkrd_sigma(J+After,I)=Bkrd_sigma(J+After,Number_peaks)
18533         Bkrd_rdgs(J+After,I)=Bkrd_rdgs(J+After,Number_peaks)
18536       NEXT J
18539     NEXT I
18542   END IF
18545   FOR J=0 TO 1  ! print bkrds & sigmas for all isotopes
18548     L=0
18551     FOR I=1 TO Mn
18554       FOR K=Number_peaks+1 TO I-1  ! but don't print redundant interf-mon. isotopes
18557         IF (Ratio_isotope(K)=Ratio_isotope(I)) OR (Ratio_isotope(I)=0) THEN 18572
18560       NEXT K
18563       M=1+After+J
18566       L=L+1
18569       IF Ratio_isotope(I) THEN PRINT TAB(L*10-9);VAL$(INT(Bkrds(M,I)))&"/"&VAL$(DROUND(Bkrd_sigma(M,I),2));
18572     NEXT I
18575     PRINT "      ";
18578     IF J=0 THEN PRINT "below, ";
18581     IF J=1 THEN PRINT "above, ";
18584     IF NOT After THEN PRINT "before pk-tops"
18587     IF After THEN PRINT "after pk-tops"
```

```
18590  NEXT J
18593  FOR I=1 TO Nn ! are above/below counts significantly different?
18596    IF Ratio_isotope(I) THEN
18599      M=1+After
18602      Qa=(FNSt((Bkrd_rdgs(M,I)))*Bkrd_sigma(M,I))^2/Bkrd_rdgs(M,I)
18605      Qb=(FNSt((Bkrd_rdgs(M+1,I)))*Bkrd_sigma(M+1,I))^2/Bkrd_rdgs(M+1,I)
18608      Qc=ABS(Bkrds(M,I)-Bkrds(M+1,I))
18611      IF NOT Mu AND Qc>1.5*SQR(Qa+Qb) THEN PRINT "STEPPED";Ratio_isotope(I);" BACKGROUNDS"
18614    END IF
18617  NEXT I
18620  SUBEXIT
18623   !
18626 Share_bkrds:Bkrds(K+After,J)=Bkrds(P,I)    ! share bakckgrounds for equivalent magnet-positions
18629  Bkrd_sigma(K+After,J)=Bkrd_sigma(P,I)+(NOT Bkrd_sigma(P,I))/10
18632  Bkrd_rdgs(K+After,J)=Bkrd_rdgs(P,I)
18635  RETURN
18638 !
18641 Bmc_out:Bmc_out=1
18644  SUBEXIT
18647 Next_run:Next_run=1
18650  SUBEXIT
18653 !
18656  Subflag=4 ! requested escape
18659 SUBEND ! ----------------------------------------------------------
18662 !
18665 !
18668 Calc_sigma:SUB Calc_sigma(Sigma,Sums,Sums_of_squares,INTEGER N)
18671  S=N*Sums_of_squares-Sums^2 ! calculate standard deviation
18674  Sigma=SQR(S*(S)0)/(N*(N-1))) ! guard against negative square-roots
18677 SUBEND ! ----------------------------------------------------------
18680   !
18683   !
18686 Invert:DEF FNInvert$(R$)   ! invert the ratio-string
18689  RETURN R$[POS(R$,"/")+1]&"/"&R$[1,POS(R$,"/")-1]
18692 FNEND ! ----------------------------------------------------------
18695 !
18698 !
18701 Getel:SUB Getel(Type,Runclass$,Nuclide$(*),Normal0(*),Norm0,Na$,INTEGER Niso,Rf,Hv0,Magnet0(*),Coarsebin(*),Ref)
18704  OPTION BASE 1
18707   !
18710  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
18713  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
18716  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
18719  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
18722  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarserange
18725  COM /Data1/ Peak(*),Pk,Interfere(*),Normal(*),T00,INTEGER Data_iso(*),Data_collector,Inv,Bmc_out,Next_run,Spike
18728  OFF KEY
18731  Subflag=0
18734  ON ERROR GOTO 18794
18737  ASSIGN @Path1 TO "TYPE:INTERNAL"  ! get element-series data from disk in right-drive
18740  ENTER @Path1,Type;Runclass$,Niso,Mcoef(*),Magnet(*),Coarsemag(*),Peak_inter,Aside,Rf,Nuclide$(*),Normal(*),Inv,Interfere(*),H
v0
18743  MAT Normal0= Normal
18746  FOR I=1 TO 24
18749    Magnet0(I)=Magnet(I,2)
18752  NEXT I
18755  OFF ERROR
18758  PRINTER IS CRT
18761  PRINT USING "4/"
18764  L=Rf  !                          Order of reference-isotope in A-list
```

```
18767  Norm0=Normal(1)  !              Normalizing isotope
18770  IF NOT Auto THEN Na$=""
18773  FOR Coarserange=0 TO 10  !      Coarse magnet-range
18776    IF Coarsebin(Coarserange)=Coarsemag(1) THEN 18782
18779  NEXT Coarserange
18782  Ref=Magnet(Rf,1) !              Reference isotope
18785  MAT Daly_ok= (0)
18788  Disp_elvals(Runclass$,Nuclide$(*),Normal(*),Interfere(*),Coarserange,Ref,Hv0,Magnet(*),Niso,Inv)
18791  SUBEXIT
18794  PRINT USING "2/,3(K),/";Ci$&" ELEMENT-SERIES ",Type," NOT DEFINED ON DISK IN RIGHT-HAND DRIVE "&Cn$
18797  Subflag=1
18800  Clunk
18803  WAIT 2
18806 SUBEND ! ---------------------------------------------------------------
18809  !
18812  !
18815 Axes:SUB Axes(Wxmin,Wxmax,Wymin,Wymax,Xmin,Xmax,Ymin,Ymax,X$,Y$,Unit_xtick,INTEGER Daly,Logplot)     ! draw plot-box for graph
ics display
18818 ! if Unit_xtick is 1, draw a tick for each integral X-axis value (used for wtd-average graphics).  If Daly is 1, add "DALY" to
 Y-axis label.
18821 ! If Daly is 0, add "CUP", & if Daly is negative, don't add nothing
18824 ! if Logplot is 1, use a logarithmic Y-axis
18827 ! Wxmin,Wxmax,Wymin,Wymax define windows of CRT (0-100)
18830  DIM Yn$[24]
18833  OFF KEY
18836  OFF KBD
18839  GINIT
18842  CSIZE 3.3
18845  DEG
18848  GCLEAR
18851  GRAPHICS ON
18854  OUTPUT KBD;CHR$(255)&CHR$(75);
18857  VIEWPORT Wxmin*RATIO,Wxmax*RATIO,Wymin,Wymax
18860  Xspred=Xmax-Xmin
18863  Yspred=Ymax-Ymin
18866  IF Unit_xtick THEN
18869    Xtick=1
18872  ELSE
18875    Xtick=DROUND(Xspred/5,1)
18878   !
18881    FOR I=1 TO 12! force lower-bound X value to have a min # of sig-figs
18884      X=FNDr(ABS(Xmin),I)
18887      IF ABS(Xmin)-X<Xtick THEN 18893
18890    NEXT I
18893    IF Xmin)=0 OR ABS(Xmin)=X THEN
18896      Xmin=SGN(Xmin)*X
18899    ELSE
18902      Xmin=-X-Xtick
18905    END IF
18908    FOR X=Xmin TO Xmax STEP Xtick  ! force Xmax to lie on tick
18911      IF X+Xtick)=Xmax THEN
18914        Xmax=X+Xtick
18917        GOTO 18932
18920      END IF
18923    NEXT X
18926  END IF
18929 !
18932  IF Logplot THEN
18935    Ytick=1
18938  ELSE
```

```
18941   Ytk=10^INT(LGT(ABS(Yspred+(NOT Yspred))))/8
18944   IF ABS(Yspred/Ytk)>12 THEN
18947     Ytk=2*Ytk
18950     GOTO 18944
18953   END IF
18956   W=ABS(Ytk)/10^INT(LGT(ABS(Ytk+(NOT Ytk))))
18959   IF W<>INT(W) THEN
18962     Ytick=INT(W)*10^INT(LGT(ABS(Ytk+(NOT Ytk))))
18965   ELSE
18968     Ytick=Ytk
18971   END IF
18974   FOR I=1 TO 12 ! force lower-bound Y to have a min # of sig-figs
18977     Y=FNDr(ABS(Ymin),I)
18980     IF ABS(ABS(Ymin)-Y)<=Ytick THEN 18986
18983   NEXT I
18986   IF Ymin>=0 OR ABS(Ymin)=Y THEN
18989     Ymin=SGN(Ymin)*Y
18992   ELSE
18995     Ymin=-Y-Ytick
18998   END IF
19001   FOR Y=Ymin TO Ymax STEP Ytick  ! force Ymax to lie on tick
19004     IF Y+Ytick>=Ymax THEN
19007       Ymax=Y+Ytick
19010       GOTO 19025
19013     END IF
19016   NEXT Y
19019 END IF
19022 !
19025 Xspred=Xmax-Xmin
19028 Yspred=Ymax-Ymin
19031 WINDOW Xmin-Xspred/8,Xmax,Ymin-Yspred/(10-(Wymax-Wymin)<60)),Ymax+Yspred/20
19034 CLIP Xmin,Xmax,Ymin,Ymax ! Use Ymin-Ytick/10 if interference
19037 AXES Xtick/(5*(NOT Unit_xtick)+Unit_xtick),Ytick,Xmin,Ymin,5-4*Unit_xtick,1,3+2*Logplot
19040 AXES Xtick/(5*(NOT Unit_xtick)+Unit_xtick),Ytick,Xmax,Ymax,5-4*Unit_xtick,1,3+2*Logplot
19043 IF Logplot THEN  ! draw extra Y-ticks
19046   FOR A=0 TO INT(Ymax-Ymin)-1
19049     FOR S=2 TO 9
19052       MOVE Xmin+Xspred/80,Ymin+A+LGT(S)
19055       IDRAW -Xspred/80,0
19058       IMOVE 79*Xspred/80,0
19061       IDRAW Xspred/80,0
19064     NEXT S
19067   NEXT A
19070 END IF
19073 CLIP OFF
19076 LORG 6
19079 FOR I=Xmin TO Xmax STEP Xtick*((NOT Unit_xtick)+Unit_xtick*(1+(Xmax>16)+2*(Xmax>30)))
19082   IF I<Xmax THEN ! don't label tick if at right-edge of box
19085     MOVE I,Ymin
19088     LABEL I
19091   END IF
19094 NEXT I
19097 E=(2-Logplot)*Ytick
19100 LORG 8
19103 FOR I=Ymin TO Ymax STEP E
19106   MOVE Xmin+(Xspred/(80-40*Logplot)),I-Logplot*(Yspred/80)*(I=Ymax)+(Yspred/70)*(I=Ymin)
19109   IF (NOT Logplot) THEN LABEL I
19112   IF Logplot THEN LABEL 10^I;" "
19115 NEXT I
19118 LORG 6
```

```
19298  IF Minmag<0 OR Minmag0<0 THEN
19301    Minmag=0
19304    Minmag0=0
19307  END IF
19310  IF Maxmag>9999 OR Maxmag0>9999 THEN
19313    Maxmag=1.E+4
19316    Maxmag0=9999
19319  END IF
19322  Max_pk=VAL(Response$(3))
19325  IF NOT Coarsechange THEN Mass_speed=VAL(Response$(4))
19328  IF Coarsechange THEN Mag_speed=VAL(Response$(4))
19331  Logscan=(UPC$(Response$(5))<>"LIN")
19334  IF Logscan AND Max_pk<10 THEN Max_pk=10
19337  Coarse=VAL(Response$(6))
19340  IF Coarse<>Coarse0 THEN
19343    Coarsechange=1
19346    Coarse0=Coarse
19349    Coarsemag(1)=Coarsebin(Coarse)
19352    OUTPUT 8 USING "4A,4Z";"$0FK",Coarsemag(1)
19355    Prompt$(1)="Start scan at magnet-value (0-9999)"
19358    Prompt$(2)="End scan at magnet-value (0-9999)"
19361    Prompt$(4)="Scan speed (magnet-units/sec)"
19364    Range(4,1)=1
19367    Range(4,2)=2000
19370    Response$(1)="0"
19373    Response$(2)="9999"
19376    Response$(4)="100"
19379    GOTO Fillout_form
19382  END IF
19385  OUTPUT 8 USING "4A,4Z";"$0FJ",Minmag
19388  IF NOT Logscan THEN
19391    Ymin=0
19394    Ymax=Max_pk
19397  ELSE
19400    M=LGT(Max_pk)
19403    Ymax=INT(M)+((M MOD INT(M))<>0)
19406    Ymin=-1-Mu-(Ymax(3)*Mu
19409  END IF
19412  ON KEY 9 LABEL "    ESCAPE" GOTO 19574
19415  ON KEY 0 LABEL "DOUBLE SPEED" GOSUB Double
19418  ON KEY 1 LABEL " HALVE SPEED" GOSUB Halve
19421  Axes(0,100,25,100,Minmag,Maxmag,Ymin,Ymax,"MAGNET UNITS","mV BEAM",0,Mu,(Logscan))
19424  Yspred=Ymax-Ymin
19427  Xspred=Maxmag-Minmag
19430  IF NOT Coarsechange THEN
19433    Mass_inter=(FNIsomag(Mcoef(*),(Niso))-FNIsomag(Mcoef(*),1))/(Niso-1)
19436    Mag_speed=Mass_speed*Mass_inter
19439    FOR I=1 TO Niso
19442      IF (Minmag<Magnet(I,2)) AND (Maxmag>Magnet(I,2)) THEN
19445        LINE TYPE 4
19448        MOVE Magnet(I,2),Ymin
19451        IDRAW 0,.91*Yspred
19454        MOVE Magnet(I,2),Ymin+Yspred*.94
19457        LINE TYPE 1
19460        IF FNEven(I) OR Xspred/Mass_inter<13 THEN
19463          LORG 5
19466          CSIZE 3
19469          LABEL Magnet(I,1)
19472        END IF
19475      END IF
```

```
19121  MOVE Xmax-Xspred/2,Ymin-Yspred*(.045+.03*((Wymax-Wymin)<60))
19124  LABEL X$
19127  LDIR 90
19130  MOVE Xmin-Xspred/8,Ymax-Yspred/2
19133  IF (Daly=0) AND (Y$()"") THEN LABEL Y$&"   (CUP)"
19136  IF Daly>0 AND (Y$()"") THEN LABEL Y$&"   (DALY)"
19139  IF Daly<0 THEN LABEL Y$
19142  LDIR 0
19145  CLIP Xmin,Xmax,Ymin,Ymax
19148 SUBEND ! ------------------------------------------------------------
19151 !
19154 !
19157 Magnet_scan:SUB Scan(Stripchart,Miniso,Maxiso,Max_pk,Mass_speed,Logscan,INTEGER Coarsebin(*),Niso,I_t)
19160  ! Scan magnet over specified range & speed
19163  OPTION BASE 1
19166  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),CoarseO
19169  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
19172  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
19175  COM /Filaments/ Filament(*),fil$(*),INTEGER Nfils,Filnum
19178  INTEGER Pr
19181  DIM Prompt$(6)[36],Response$(6)[50],Use(6),Range(6,2),P1(2),P2(2)
19184  DATA "Start scan at isotope","End scan at isotope","Max. beam on graph (mV)","Scan-speed (mass-units/second)","Linear or Log
Scan (Lin/Log)"
19187  DATA"Coarse-magnet range (0-10)"
19190  DATA "","","","","","",0,9990,10,9999,.01,10000,.02,10,-1,-1,0,10
19193  OFF KEY
19196  OFF KNOB
19199  Coarse_in=CoarseO
19202  READ Prompt$(*),Response$(*),Range(*)
19205  MAT Use= (1)
19208  Response$(1)=VAL$(Miniso)
19211  Response$(2)=VAL$(Maxiso)
19214  Response$(3)=VAL$(Max_pk)
19217  Response$(4)=VAL$(Mass_speed)
19220  IF Logscan THEN Response$(5)="LOG"
19223  IF NOT Logscan THEN Response$(5)="LIN"
19226  !
19229 Fillout_form:  !
19232  Response$(6)=VAL$(CoarseO)
19235  Form(Prompt$(*),Response$(*),Use(*),Range(*),6,1,Escape,3)
19238  IF Escape THEN 19574
19241  IF NOT Coarsechange THEN
19244    Miniso=VAL(Response$(1))
19247    Maxiso=VAL(Response$(2))
19250    Minmag=FNIsomag(Mcoef(*),(Miniso))
19253    Maxmag=FNIsomag(Mcoef(*),(Maxiso))
19256  ELSE
19259    Minmag=VAL(Response$(1))
19262    Maxmag=VAL(Response$(2))
19265  END IF
19268  MinmagO=Minmag
19271  MaxmagO=Maxmag
19274  Magspan=Maxmag-Minmag
19277  IF Magspan>300 THEN
19280    Minmag=100*INT(Minmag/100)   ! need somewhat rounded plot-boundaries
19283    Maxmag=100*INT(Maxmag/100)+100
19286  ELSE
19289    Minmag=INT(Minmag)
19292    Maxmag=INT(Maxmag)+1
19295  END IF
```

```
19658  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
19661  INTEGER Pr
19664  OFF KNOB
19667  ON KEY 9 LABEL "    ESCAPE" GOTO 19769
19670  D=INT(.80*Rside)
19673  Max_inc=.01
19676  X=Magnet(L,2)-D
19679  A=X
19682  C=Magnet(L,2)+D
19685  Ymin=DROUND(Peaksize*.92,2)
19688 Start_graph:Ymax=Peaksize*(1+Max_inc)
19691  Yspred=Ymax-Ymin
19694  OUTPUT 8 USING "4A,4Z";"$OFJ",X
19697  Axes(D,100,25,100,-D,D,Ymin,Ymax,"MAGNET","mV",0,Mu,0)
19700  CSIZE 3.3
19703  LORG 5
19706  MOVE 0,Ymin+Yspred/10
19709  LABEL TRIM$(Nuclide$(L)&"-"&VAL$(Magnet(L,1));"  PEAK-SHAPE"
19712  MOVE 0,Ymax-Yspred/10
19715  LABEL "SCAN UP"
19718  FOR P=1 TO -1 STEP -2
19721    MOVE -D,Peaksize/2
19724    FOR I=X TO (P>0)*C+(P<0)*A STEP P
19727      OUTPUT 8 USING "4A,4Z";"$OFJ",I
19730      Enter_beam(C,Mv,L,I_t,Pr)
19733      IF Mv>Ymax THEN  ! exceeds Y-max of graph- increase Ymax & redo
19736        Max_inc=Max_inc+.03
19739        GOTO Start_graph
19742      END IF
19745      IF P>0 THEN DRAW I-Magnet(L,2),Mv
19748      IF P<0 THEN DRAW Magnet(L,2)-I+0,.99*Mv
19751    NEXT I
19754    X=C
19757    MOVE 0,Ymin+Yspred/4
19760    LABEL "SCAN DOWN"
19763    MOVE 0,Mv
19766  NEXT P
19769 SUBEND ! ----------------------------------------------------------
19772 !
19775 !
19778 Avgraf:SUB Avgraf(Ratio(*),N,R$,REAL Percent_error(*),Time(*))  ! weighted-average graphics
19781  OPTION BASE 1
19784  DIM Err(80)
19787  FOR I=1 TO N
19790    Err(I)=2*Ratio(I)*Percent_error(I)/100
19793  NEXT I
19796  Ymax=-9.E+99
19799  Ymin=+9.E+99
19802  Boxwidth=2.5
19805  FOR I=1 TO N
19808    IF Ymax<Ratio(I)+Err(I) THEN Ymax=Ratio(I)+Err(I)
19811    IF Ymin>Ratio(I)-Err(I) THEN Ymin=Ratio(I)-Err(I)
19814  NEXT I
19817  Yspred=ABS(Ymax-Ymin)
19820  Xmin=Time(1)-(Time(N)-Time(1))/(5+3*(N>5))
19823  IF Xmin<0 THEN Xmin=0
19826  Xmax=Time(N)+Boxwidth/2
19829  Xspred=Xmax-Xmin
19832  Axes(0,100,45,100,Xmin,Xmax,Ymin-Yspred/10,Ymax+Yspred/10,"TIME (minutes)  ["&R$&"]",R$,0,-1,0)
19835  CSIZE 3.3
```

```
19478    NEXT I
19481  END IF
19484  LINE TYPE 1
19487  Ex_ymin=Logscan*10^Ymin
19490  MOVE Minmag,0
19493  A=INT(Mag_speed/(4*I_t-3))
19496  C=0
19499  Started=0
19502  P=A+(NOT A)
19505  P2(1)=Minmag0-P/2
19508  IMAGE "MAGNET: ",4D,10X,"SCAN-SPEED: ",4D,10X,"mV BEAM: ",K
19511  IMAGE "NUCLIDE:  ",3D.D," ",5X,"MAGNET: ",4D,5X,"SCAN-SPEED: ",3D.2D,7X,"mV BEAM: ",K
19514  WHILE P2(1)<=Maxmag0
19517    Mpos=INT(P2(1)+P/2)
19520    IF Mpos>9999 THEN Mpos=9999
19523    OUTPUT 8 USING "4A,4Z";"$0FJ",Mpos
19526    Enter_beam(0,Mv,1,I_t,Pr)
19529    IF Coarsechange THEN DISP USING 19508;P2(1),Mag_speed,DROUND(Mv,3)
19532    IF NOT Coarsechange THEN DISP USING 19511;PROUND(FNMag_iso(Mcoef(*),(P2(1))),-1),P2(1),Mass_speed,DROUND(Mv,3)
19535    IF Pr>1 THEN 19526
19538    IF NOT Logscan THEN
19541      Mv=Mv+Yspred/200
19544      P2(2)=Mv
19547    ELSE
19550      IF Mv<=Ex_ymin THEN Mv=1.05*Ex_ymin
19553      P2(2)=LGT(Mv)
19556    END IF
19559    IF Started THEN CALL Thickpen(P1(*),P2(*),Yspred/150,1)
19562    MAT P1= P2
19565    P2(1)=P+P2(1)
19568    Started=1
19571  END WHILE
19574  MOVE 0,0
19577  Stripchart=0
19580  Coarse0=Coarse_in
19583  Coarsemag(1)=Coarsebin(Coarse0)
19586  SUBEXIT
19589 Double:IF Mag_speed<2000 THEN
19592    P=P*2
19595    BEEP 1000,.1
19598    Mag_speed=Mag_speed*2
19601    Mass_speed=Mass_speed*2
19604  END IF
19607  RETURN
19610 Halve:IF Mag_speed>1 THEN
19613    P=P/2
19616    Mag_speed=Mag_speed/2
19619    Mass_speed=Mass_speed/2
19622    BEEP 100,.1
19625  END IF
19628  RETURN
19631 SUBEND ! ----------------------------------------------------------------
19634 !
19637 !
19640 Peakshape:SUB Shape(Peaksize,Nuclide$(*),INTEGER I_t)
19643 ! look at peakshape graphics
19646  OPTION BASE 1
19649  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
19652  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
19655  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
```

```
19838  AREA COLOR .5,.5,.5
19841  FOR I=1 TO N
19844    MOVE Time(I)-Boxwidth/2,Ratio(I)-Err(I)
19847    RECTANGLE Boxwidth,2*Err(I),FILL,EDGE
19850  NEXT I
19853 SUBEND ! ------------------------------------------------------------
19856  !
19859  !
19862 Flag:SUB Flag(Flag(*),INTEGER Flag_mu,OPTIONAL Noprint)      ! check filament flags
19865  OPTION BASE 1
19868  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
19871  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
19874  DIM F$(4)[20]
19877  DATA "NO FILAMENTS","SINGLE-FILAMENT","SIDE-FILAMENTS ONLY","TRIPLE-FILAMENT"
19880  READ F$(*)
19883  FOR I=2 TO 1 STEP -1
19886    OUTPUT 8;"$OMW"&VAL$(I-1)&":28"
19889    OUTPUT 8;"$IMW"
19892    ENTER 8;F
19895    Flag(I)=F
19898    IF NPAR=2 THEN
19901      IF I=2 THEN PRINT USING "2/,K,#";"PREHEAT-SAMPLE:  "
19904      IF I=1 THEN PRINT "RUNNING-SAMPLE:  ";
19907      PRINT USING "K,/";FNH$(F$(F+1))
19910      IF ((I=1) AND (F=0)) OR (F=2) THEN CALL Clunk
19913    END IF
19916    OUTPUT 8;Mm$(Flag_mu,I_t)
19919  NEXT I
19922 SUBEND ! ------------------------------------------------------------
19925 !
19928 !
19931 Cat:SUB Cat(INTEGER Prtr(*))    ! Print Run-Directory
19934  DIM Date$[12],Na$[50],N$[10],S$[80],R$(10)[60]
19937  INTEGER Sample
19940  OFF KEY
19943  OFF KNOB
19946  MAT R$= ("")
19949  R$(1)="SEARCH THE DATA-DISK FOR A RUN WITH A PARTICULAR SAMPLE-NAME"
19952  R$(2)="SHOW THE DATA-DISK DIRECTORY FOR RUNS 1 to 32"
19955  R$(4)="DISPLAY THE DISK-INFORMATION ON THE CRT ONLY (default)"
19958  R$(5)="PRINTOUT  '   '      '      '   ' PRINTER"
19961  R$(10)="RETURN TO THE BMC"
19964  CALL Keymenu(R$(*))
19967  ON KEY 0 LABEL " SAMPLE-NAME" GOTO 20009
19970  ON KEY 1 LABEL "  DIRECTORY" GOTO 20159
19973  ON KEY 3 LABEL "  CRT DISPLAY" GOSUB Crt
19976  ON KEY 4 LABEL "   PRINTOUT" GOSUB Printout
19979  ON KEY 9 LABEL "    ESCAPE" GOTO 20249
19982  GOTO 19982
19985  !
19988 Crt:Printout=0
19991  PRINT "CRT DISPLAY ONLY"
19994  RETURN
19997 Printout:Printout=1
20000  PRINT "PRINTOUT SEARCH ON PRINTER"
20003  RETURN
20006  !
20009  OFF KEY
20012  S$=""
20015  PRINT USING "18/,K,/,K,/";"ENTER THE CHARACTERS THAT YOU WANT MATCHED IN THE FIRST 10 CHARACTERS","OF THE RUN-NAME (or pres
```

```
20192    NEXT P
20195  NEXT I
20198  IF NOT Printout THEN
20201    DISP "PRESS  CONTINUE  WHEN READY TO GO ON, USE ARROWS TO SCROLL DISPLAY."
20204    PAUSE
20207  END IF
20210  GOTO 19946
20213  !
20216 Retrn:GOTO 20132
20219  PRINTER IS 1
20222  PRINT USING "K,K";"CAN'T ACCESS DATA FOR RUN# ",I
20225  RETURN
20228  PRINTER IS 1
20231  PRINT USING "2/,K";"UNABLE TO ACCESS THE RESDIR FILE ON THE LEFT-HAND DISK."
20234  GOTO 20243
20237  PRINTER IS 1
20240  PRINT USING "2/,K";"UNABLE TO ACCESS THE RESULT FILE ON THE LEFT-HAND DISK."
20243  PRINT USING "2/,K";"PRESS CONTINUE TO RETURN TO BMC."
20246  PAUSE
20249 SUBEND ! ------------------------------------------------------------
20252 !
20255 Focshift:SUB Focshift(INTEGER Foc(*))
20258  PRINT USING "18/,K,2X,8(3D,3X)";"FOCUS SETTINGS:",1,2,3,4,5,6,7,8
20261  PRINT USING "18X,8(3D,3X),5/";Foc(*)
20264  OFF KEY
20267  OFF KBD
20270  OFF KNOB
20273  LOOP
20276    Plate=0
20279    LOOP
20282      INPUT "PLATE#, NEW VALUE (0-999)?  (press CONT to escape)",Plate,F
20285    EXIT IF (Plate(8)*(F)=0 AND F(=999) OR (Plate=8)*(F)=460 AND F(=540)
20288      Clunk
20291    END LOOP
20294  EXIT IF Plate(1 OR Plate)8
20297    Foc(Plate)=F
20300    PRINT TABXY(18+(Plate-1)*6,13);FNH$(VAL$(F))&"  "
20303  END LOOP
20306 SUBEND ! ------------------------------------------------------------
20309 !
20312 !
20315 Find:SUB Find(INTEGER Redo,Barnum,Nfils,Subflag,REAL Filament(*),Time0)
20318  OPTION BASE 1
20321  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
20324  COM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel
20327  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
20330  ! Rotate a new sample into position
20333  DIM Fil_flags(2)
20336  OFF KEY
20339  OFF KNOB
20342  MAT Daly_ok= (0)
20345  Default_barrel=81+187*(FNBarpos((Barnum))-1) ! Rough estimate of correct barrel position for this sample
20348  Mu=0
20351  Subflag=0
20354  MAT Filament= (0)
20357  OUTPUT 8;"$OMW08000"
20360  Zero_fils(Filament(*))
20363  IF Redo=0 THEN PRINT USING "3(K),/";"ROTATING BARREL# ",Barnum," INTO POSITION  (default position "&VAL$(Default_barrel)&")"
20366  Bct=1 ! try# counter
20369  !
```

```
20372  BarposO=Default_barrel
20375  Barrel_posO=Default_barrel
20378  Barrel_position=Default_barrel-70
20381  Barrel_position=Barrel_position*(Barrel_position>=0)
20384  Resbar
20387  DISP CHR$(130)&"ROTATING TO BARREL# ";Barnum;"  -- TARGET: ":Barrel_position;CHR$(128)
20390  CALL Brl(Barrel_position,F,1,0)
20393  OUTPUT 8;"$OMW9??8"
20396  OUTPUT 8;"$IMW"
20399  ENTER 8;E
20402  IF E THEN
20405    WAIT .2
20408    GOTO 20393
20411  END IF
20414  !
20417  OUTPUT 8;"$OM00200$OM10200"   !  put 0.2 amps through the sample filaments to check the filament flags
20420  WAIT .1
20423  !
20426  LOOP
20429    Barrel_position=Barrel_position+1
20432    CALL Brl(Barrel_position,F,.09) ! Rotate barrel until get filament contact
20435  EXIT IF FNFiltest(Nfils,F) OR (Barrel_position-Default_barrel)>70) ! found filament-assembly
20438  END LOOP
20441  !
20444  IF Barrel_position-Default_barrel)>70 THEN Default_exit
20447  !
20450  Min_barrel=Barrel_position   ! Min_barrel is the position where contact is just made (positive rotation direction!)
20453  REPEAT
20456    Barrel_position=Barrel_position+1
20459    Barrel_delta=Barrel_position-Default_barrel
20462    CALL Brl(Barrel_position,F,.09) ! rotate in a positive direction until contact is lost
20465  UNTIL FNFiltest(Nfils,F)=0 OR Barrel_delta)70
20468  IF Barrel_delta)70 THEN
20471    Subflag=1
20474    PRINT USING "3/,3(K),2/";Ci$&" OPEN CIRCUIT IN BARREL# ",Barnum," "&Cn$&"   (Is Filament Supply "&FNB1$("ON")&FNH$("??)")
20477    GOTO Default_exit
20480  END IF
20483  !
20486  Max_barrel=Barrel_position-1   ! Max_barrel is the barrel-position where filament contact is just lost (positive rotation!)
20489  Barrel_position=(Max_barrel+Min_barrel)/2
20492  IF ABS(Barrel_posO-Barrel_position))34 OR Max_barrel-Min_barrel<9 THEN
20495  ! shift the default position if the "best" pos'n differs from the default pos'n by more than 35 units.
20498 Default_exit:Barrel_position=Default_barrel    ! default barrel-value
20501    Max_barrel=Barrel_position+8 ! default 'above' barrel-value
20504    Min_barrel=Barrel_position-8 ! default 'below' barrel-value
20507    PRINT FNB1$("***USING DEFAULT BARREL-VALUES***")&"     (should check manually)"
20510    Subflag=2
20513    Beep(160,.05,.03,30)
20516  END IF
20519  !
20522  Barrel_posO=Barrel_position
20525  Barrel_position=(Barrel_position-70)*(Barrel_position>70)
20528  CALL Brl(Barrel_position,F,1)   ! rotate barrel to the final position
20531  !
20534  REPEAT
20537    CALL Brl(Barrel_position+1,F1,.09)
20540    Barrel_position=Barrel_position+1
20543  UNTIL Barrel_position=Barrel_posO
20546  !
20549 Found_sample:PRINTER IS CRT
```

```
s CONTINUE to escape)."
20018  PRINT USING "K,/,K,/";"ANALYST WILL THEN SEARCH THE DATA-DISK FOR ALL RUNS WITH THOSE CHARACTERS,","AND SHOW WHICH FILE-NUM
BERS THEY ARE LISTED UNDER."
20021  PRINT USING "K,/,K";"YOU CAN THEN USE THESE FILE-NUMBERS TO ACCESS DATA-BLOCKS WHOSE DIRECTORY","HAS BEEN OVER-WRITTEN."
20024  INPUT S$
20027  IF S$="" THEN 19946
20030  S$=UPC$(TRIM$(S$))
20033  DISP
20036  ON ERROR GOTO 20237
20039  ASSIGN @Path1 TO "RESULT:INTERNAL,4,1"
20042  OUTPUT KBD;CHR$(255)&CHR$(75);
20045  ON ERROR GOTO Retrn
20048  Fi1=0
20051  ON KEY 9 LABEL "    ESCAPE" GOTO 19946
20054  FOR I=1 TO 500
20057    ENTER @Path1,I;N$[1,10]
20060    PRINT I;TAB(6);N$
20063    IF POS(UPC$(N$),S$) THEN
20066      IF Fi1=0 THEN Fi1=I
20069      Fi2=I
20072    ELSE
20075      IF Fi1 THEN
20078        FOR P=1+Printout TO 1 STEP -1
20081          PRINTER IS Prtr(P)
20084          PRINT USING "3/,2(K,X),2(K),/";"FOUND RUN WITH "&CHR$(34)&S$&CHR$(34)&" IN FILES ",Fi1,"TO ",Fi2
20087        NEXT P
20090        DISP "CONTINUE OR QUIT SEARCHING FOR MORE "&CHR$(34)&S$&CHR$(34)&"?"
20093        OFF KEY
20096        ON KEY 0 LABEL "   CONTINUE" GOTO 20108
20099        ON KEY 4 LABEL "     QUIT" GOTO 19946
20102        GOTO 20102
20105        !
20108        Fi1=0
20111        Fi2=0
20114        DISP
20117        OFF KEY
20120        ON KEY 9 LABEL "    ESCAPE" GOTO 19946
20123        GOTO 20132
20126      END IF
20129    END IF
20132  NEXT I
20135  OFF ERROR
20138  OFF KEY
20141  DISP "NO SUCH RUN-NAME EXISTS ON THE DISK.  DO YOU WANT TO TRY AGAIN (Y/N)?"
20144  OFF KEY
20147  ON KEY 0 LABEL "    YES" GOTO 20009
20150  ON KEY 4 LABEL "     NO" GOTO 20249
20153  GOTO 20153
20156  !
20159  OFF KEY
20162  ON KEY 9 LABEL "    ESCAPE" GOTO 19946
20165  OUTPUT KBD;CHR$(255)&CHR$(75);
20168  ON ERROR GOTO 20231
20171  ASSIGN @Path1 TO "RESDIR:INTERNAL,4,1"
20174  ON ERROR GOTO 20195
20177  FOR I=1 TO 32
20180    ENTER @Path1,I;Sample,Date$,Na$,Fi1,Fi2
20183    FOR P=Printout+1 TO 1 STEP -1
20186      PRINTER IS Prtr(P)
20189      PRINT USING "8A,2D,5X,5A,2D,8X,12A,8X,4(K),/,K";"BARREL# ",Sample,"RUN# ",I,Date$,"File #s ",Fi1," to ",Fi2,Na$
```

```
20552  OUTPUT 8;"$OP00200$OP10200" ! put .2 amps through preheat filaments
20555  IMAGE /,3(K),15X,3(4D,3X),/,3(K)
20558  PRINT USING 20555;"BARREL# ",Barnum," IN POSITION",Min_barrel,Barrel_position,Max_barrel,"CONTACT-WIDTH IS ",Max_barrel-Min_b
arrel," BARREL-UNITS"
20561  IF Max_barrel-Min_barrel<15 THEN PRINT "(this amount of contact is marginal  take care)"
20564  CALL Flag(Fil_flags(*),0)
20567  Zero_fils(Filament(*))
20570  Time0=TIMEDATE  !  Time0 is # of seconds since sample was rotated into position
20573 SUBEND ! ----------------------------------------------------------
20576 !
20579 !
20582 DEF FNDiff(Peak1,Peak2,Dark_var,INTEGER I_t,REAL Ions)
20585 ! uncertainty in beamsize difference, from resistor & counting noise
20588  T=1/(4*I_t-3) ! integration time in seconds
20591  RETURN 2.2*SQR((2*Dark_var+(Peak1+Peak2)/Ions)/T)
20594 FNEND! ------------------------------------------------------------
20597 !
20600 !
20603 Limit:DEF FNLimit(X,L1,L2)  ! (for FOCUS) Is value within focus-limit bounds?
20606  RETURN (X<L1) OR (X>L2)
20609 FNEND! ------------------------------------------------------------
20612  !
20615  !
20618 Fnst:DEF FNSt(N)  ! Student's-t approximation (N-1 d.f.) (from Andy Turek)
20621  RETURN 12.7*(N=2)+(N>2)*1.96*(N-1)/SQR((N-1)^2-2.43*(N-1)+1.696)
20624 FNEND! ------------------------------------------------------------
20627!
20630!
20633 Enter_beam:SUB Enter_beam(REAL Counts,Mv,INTEGER L,I_t,Pr,OPTIONAL REAL Wait,INTEGER Dropouts,Bmc)
20636  ! get counts from the DVM of the spectrometer, convert to millivolts, protect against oversize beams
20639  OPTION BASE 1
20642  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
20645  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
20648  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
20651  Bad_counts=0
20654  IF NPAR<8 THEN Turndown_ok=1
20657  IF NPAR=8 THEN Turndown_ok=NOT Bmc
20660 !
20663  IF NPAR>5 THEN        ! set integration time & collector
20666    OUTPUT 8;Mm$(Mu,I_t)
20669    IF Wait THEN WAIT Wait
20672  END IF
20675  !
20678  LOOP
20681    OUTPUT 8;"$IDV"     ! query DVM
20684    ENTER 8;Counts      ! Counts per integration time
20687  EXIT IF Bad_counts=3 OR Counts*(1+4*(I_t=2)))>(Ze(Mu)-150) ! guard against GPIO dropouts- don't accept any counts less than 15
0 below accepted zero
20690    Bad_counts=1+Bad_counts
20693    IF NPAR=7 THEN Dropouts=1+Dropouts
20696  END LOOP
20699  !
20702  Mv=((4*I_t-3)*Counts-Ze(Mu))/Mx(Mu)  ! millivolts beam
20705 !
20708 ! Check for oversize beam (>50 mV on Daly or >10v on Cup).
20711  Pr=1
20714  IF (Mu=0 AND Mv<1.E+4) OR (Mu AND Mv<50) THEN SUBEXIT
20717 !
20720  IF Mu=0 AND Mv>=1.E+4 THEN        ! Faraday cup beam must be <10 v.
20723    Pr=3
```

```
20726    IF Turndown_ok THEN            ! If not called from the BMC, then
20729                                   ! reduce sample-filament current by 3%,
20732      F=Nfils*(L)0)+(NOT L)        ! but turn down the center-fil if on
20735      IF F=2 AND Filament(2)=0 THEN F=1! Re-187 but a non-Re run.
20738      Target=.97*Filament(F)       ! isn't any current in the sides, turn
20741      M$="$OM0$OM1"                ! down the center filament.
20744      OFF KEY
20747      ON KEY 9 LABEL "    ESCAPE" GOTO 20819
20750      WHILE Filament(F))Target
20753        DISP " BEAM)10u:   REDUCING FILAMENT-CURRENT BY 3% - ";Filament(F),"(PRESS  k9  TO ESCAPE)"
20756        OUTPUT 8 USING "4A,4Z";Ff$(0,F),FNF(Filament(F)-.001)
20759        Filament(F)=Filament(F)-.001
20762        WAIT .02
20765      END WHILE
20768      DISP
20771    END IF
20774  END IF
20777 !
20780  IF Mu AND Mv)=50 THEN          ! Daly beam must be <50 mV
20783 ! switch from Daly to Faraday cup, wait 6 seconds
20786    Mu=0
20789    Daly_ok(L)=0
20792    Pr=2
20795    OUTPUT 8;Mm$(0,I_t)
20798    DISP USING "/,K";"BEAM TOO INTENSE FOR DALY- NOW ON CUP (6 seconds before beam will reappear)"
20801    Beep(400,.03,.02,20)
20804    WAIT 5.7
20807    DISP
20810    SUBEXIT
20813  END IF
20816   !
20819 SUBEND   ! ----------  -------  --------  ----------------------------------------
20822!
20825!
20828 Clockset:SUB Clockset
20831  ON ERROR GOTO Bad_date
20834  A$=""
20837  DISP "Enter the current date, in the format  "&DATE$(TIMEDATE);
20840  IF NOT POS(Date$,"1900") THEN DISP " ("&FNH$("CONTINUE")&" if OK)";
20843  LINPUT A$
20846  IF A$="" THEN 20858
20849  SET TIMEDATE DATE(A$)
20852  A$=DATE$(TIMEDATE)
20855  IF VAL(A$[8])<1984 OR VAL(A$[8])>2000 THEN Bad_date
20858  ON ERROR GOTO Bad_clock
20861  A$=""
20864  DISP "Enter the correct time, in the format "&FNClock_12$(TIME$(TIMEDATE)>&"   ["&FNH$("CONTINUE")&" if OK]";
20867  LINPUT A$
20870  IF A$="" THEN SUBEXIT
20873  C=POS(A$,":")
20876  A=POS(UPC$(A$),"A")
20879  P=POS(UPC$(A$),"P")
20882  H=VAL(A$)
20885  IF C=0 OR (A=0 AND P=0) THEN Bad_clock
20888  H=H+12*(P AND H<>12)-12*(A AND H=12)
20891  M=VAL(A$[1+C])
20894  IF H=24 AND M=0 THEN H=0
20897  SET TIME TIME(VAL$(H)&":"&VAL$(M))
20900  OFF ERROR
20903  SUBEXIT
```

```
20906  !
20909 Bad_date:Clunk
20912  PRINT A$;"  isn't in the requested format. Please try again."
20915  GOTO 20834
20918 Bad_clock:Clunk
20921  PRINT A$;"  isn't in the requested format. Please try again."
20924  GOTO 20861
20927 SUBEND ! ------------------------------------------------------------
20930 !
20933 !
20936 Clunk:SUB Clunk ! error-sound
20939  BEEP 220,.1
20942  BEEP 100,.2
20945 SUBEND ! ------------------------------------------------------------
20948 !
20951 !
20954 Whoop:SUB Whoop  ! up-beat sound indicating process finished
20957  OPTION BASE 1
20960  DIM T(12,2)
20963  DATA 244,325,407,488,570,730,900,980,1140,1380,1630,0
20966  MAT T= (.01)
20969  FOR I=1 TO 12
20972    READ T(I,1)
20975  NEXT I
20978  Beep(0,0,0,0,T(*))
20981 SUBEND ! ------------------------------------------------------------
20984 !
20987 !
20990 Superclunk:SUB Superclunk  ! something-is-very-wrong sound
20993  OPTION BASE 1
20996  DIM T(41,2)
20999  MAT T= (.04)
21002  T(41,1)=0
21005  FOR J=0 TO 1
21008    FOR I=1 TO 20
21011      T(I+20*J,1)=1000-I*48
21014    NEXT I
21017  NEXT J
21020  Beep(0,0,0,0,T(*))
21023 SUBEND  ! ------------------------------------------------------------
21026  !
21029  !
21032 Brl:SUB Brl(INTEGER Barrel_pos,REAL Flag,Wait,OPTIONAL Display)
21035      ! rotate barrel to Barrel_pos, wait Wait seconds, then check filament
21038      ! flags.
21041      !
21044  DATA "(no filaments)","(single filament)","(side-filaments only)","(triple filament)"
21047  DIM F$(3)[21]
21050  READ F$(*)
21053  OUTPUT 8 USING "4A,4Z";"$OBB",Barrel_pos
21056  WAIT Wait
21059  IF Flag>=0 THEN ! query filament-flags if Flag variable not negative
21062    OUTPUT 8;"$OMW0:28"
21065    OUTPUT 8;"$IMW"
21068    ENTER 8;Flag
21071    IF NPAR<4 THEN DISP "BARREL:";Barrel_pos;TAB(20);F$(Flag)
21074  END IF
21077 SUBEND ! ------------------------------------------------------------
21080      !
21083      !
```

```
21086 Resbar:SUB Resbar !reset barrel to microswitch (old UG routine?)
21089  OUTPUT 8;"$OM00000$OM10000$OP00000$OP10000"          ! turn off all fils
21092  DISP CHR$(130);"RESETTING BARREL";CHR$(128)
21095  OUTPUT 8;"$OMW0?:8"
21098  REPEAT
21101    OUTPUT 8;"$OBB8000"
21104    WAIT 1
21107    OUTPUT 8;"$OBB0000"
21110    OUTPUT 8;"$OMW8?0"
21113    WAIT .4
21116    OUTPUT 8;"$IMW"
21119    ENTER 8;Barrel_stopped
21122  UNTIL Barrel_stopped
21125  DISP
21128 SUBEND! ------------------------------------------------------------
21131 !
21134 !
21137 Zero_fils:SUB Zero_fils(Filament(*))  ! zero all filament currents
21140  OUTPUT 8;"$OM00000$OM10000$OP00000$OP10000"
21143  MAT Filament= (0)
21146 SUBEND! ------------------------------------------------------------
21149 !
21152 !
21155 Parse:SUB Parse(Input_string$,Numeric_value(*),Ninputs)
21158  ! parse a string to get embedded numeric values, separated by commas
21161  MAT Numeric_value= (0)
21164  Comma=1
21167  Ninputs=0
21170  ON ERROR GOTO Done
21173  WHILE Comma>0
21176    Numeric_value(1+Ninputs)=VAL(Input_string$)
21179    Ninputs=1+Ninputs
21182    Comma=POS(Input_string$,",")
21185    Input_string$=Input_string$[1+Comma]
21188  END WHILE
21191  !
21194 Done:SUBEND ! ------------------------------------------------------
21197 !
21200 !
21203 Keymenu:SUB Keymenu(R$(*))
21206 ! Puts a relatively detailed explanation of the softkey functions on the CRT
21209  OPTION BASE 1
21212  OUTPUT KBD;CHR$(255)&CHR$(75);
21215  PRINT TABXY(4,8);"KEY#";TAB(16);"FUNCTION"
21218  PRINT
21221  FOR I=1 TO 10
21224    IF R$(I)<>"" THEN PRINT TAB(4);I-1;TAB(7);"---";TAB(11);R$(I)
21227  NEXT I
21230 SUBEND! ------------------------------------------------------------
21233!
21236!
21239 Kybrd:SUB Kybrd(Kybrd$,K,OPTIONAL Key$) ! Retrieve keystroke from KBD$ buffer
21242  K$[1,3]=Kybrd$
21245  K=NUM(K$[1,1])
21248  IF K=255 THEN K=NUM(K$[2,2])-256    ! second byte of non-ASCII keys
21251  IF K=-1 THEN K=NUM(K$[3,3])-512    ! third  byte of non-ASCII keys
21254  IF NPAR=3 THEN Key$=CHR$(K)
21257 SUBEND! ------------------------------------------------------------
21260!
21263!
```

```
21266 Yes:DEF FNYes(Input$)
21269! return 1 for input of Y, 0 for N, -1 for space, -2 for any other input
21272! - looks at only first character of input -
21275  DIM Temp$[1]
21278  Temp$[1,1]=UPC$(TRIM$(Input$))
21281  IF Temp$="Y" THEN RETURN 1
21284  IF Temp$="N" THEN RETURN 0
21287  IF Temp$=" " THEN RETURN -1
21290  RETURN -2
21293 FNEND! ------------------------------------------------- ----------
21296!
21299!
21302 Form:SUB Form(Prompt$(*),Response$(*),Use(*),Range(*),Nchoices,Firstchoice,Escape,OPTIONAL Helpsub)
21305 ! 2/28/84
21308 ! General-purpose form fill-out screen.
21311 ! The Prompt$ array contains the prompts or labels of the parameters,
21314 ! the Response$ array as passed to the subprogram contains the default
21317 ! responses, and returns any changed responses.
21320 ! The Use array determines whether a given parameter can be accessed and
21323 ! changed.
21326 ! The Range array contains the minimum & maximum permissible values for a
21329 ! parameter (0,0 for any range,-1,-1 for any string, -2,-2 for Y/N).
21332 ! Parameters whose values must be freshly entered must be indicated with a
21335 ! passed response of "??".
21338 ! Prompt$ and Response$ must be dimensioned for string-lengths of 36 & 37-50.
21341 ! No more than 15 choices can fit on the screen at one time.
21344 ! The cursor is controlled by either the KNOB, by CONTINUE (increment parameter)/CONTROL CONTINUE
21347 ! (decrement parameter) or by the UP/DOWN arrows.
21350  DIM A$[50],B$[38],C$[50]
21353  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
21356  Escape=0
21359  B$=RPT$(" ",38)
21362  C$=RPT$(" ",50)
21365  Q$=Ci$&" "&Cn$
21368  OFF KEY
21371 Begin:CONTROL 1,4;0          ! HOME key
21374  CONTROL KBD;1               ! put CAPS LOCK on
21377  Capslock=1
21380  OUTPUT KBD;Clear$;
21383  PRINT TAB(3);FNUn$("PARAMETER");TAB(43);FNUn$("RESPONSE")
21386  FOR I=1 TO Nchoices
21389    PRINT TABXY(1,I+2);Prompt$(I);TAB(42);Response$(I)
21392  NEXT I
21395  PRINT TABXY(1,18);"Use  ARROWS, (CTRL) CONTINUE  or KNOB to move cursor";
21398  IF NPAR<0 THEN PRINT " to different parameters."
21401  IF NPAR=0 THEN PRINT ", CTRL-H(elp) for help."
21404  IF NOT Helped THEN
21407    FOR P=Firstchoice TO Nchoices
21410      IF Use(P) THEN 21416
21413    NEXT P
21416    Np=P! "new" parameter#
21419  END IF
21422  Helped=0
21425  !
21428 Change_params:IF Np>Nchoices THEN Np=1
21431  IF Np<1 THEN Np=Nchoices
21434  IF Use(Np)=0 THEN              ! goto next param if on a forbidden one
21437    Np=Np+(Np)P)-(Np(P)
21440    GOTO 21428
21443  END IF
```

```
21446  PRINT TABXY(42,P+2);B$ ! blank out response area
21449  PRINT TABXY(1,P+2);Prompt$(P);TAB(42);Response$(P)[1,38] ! restore display-format for previous parameter
21452  L1=LEN(Prompt$(Np))     ! for adding pointer-dashes
21455  PRINT TABXY(2+L1,Np+2);RPT$("-",38-L1) ! add dash-pointer
21458 Inverse:PRINT TABXY(41,Np+2);FNH$(Response$(Np)[1,37]) ! print response to be changed in inverse video
21461 !
21464  DISP " ENTER  new value   (press EXECUTE  when all parameters defined,  k9  to escape)"
21467  J=0    ! counter for position in response-string
21470  A$=Response$(Np)
21473 Get_keystroke:ON KBD ALL GOTO 21509
21476  ON KNOB .05 GOTO 21485
21479  GOTO 21479
21482    !
21485  IF KNOBX>0 THEN
21488    ON KNOB .1 GOSUB Retrn   ! dummy
21491    GOTO 21554
21494  ELSE
21497    ON KNOB .1 GOSUB Retrn   ! dummy
21500    GOTO 21575
21503  END IF
21506    !
21509  CALL Kybrd(KBD$,K)
21512  ON KBD ALL GOSUB Retrn    ! dummy
21515!
21518  SELECT K
21521  CASE -199 ! k9 key pressed - escape.
21524    Escape=1
21527    OUTPUT KBD;Clear$;
21530    GOTO 21962
21533  CASE -187                 ! ENTER key pressed
21536    Exec=0
21539    GOTO 21770
21542  CASE -168                 ! EXECUTE key pressed
21545    Exec=1
21548    GOTO 21770
21551  CASE -189,-170            ! CONTINUE or DOWN-ARROW key pressed
21554    P=Np
21557    REPEAT
21560      Np=Np+1
21563      IF Np>Nchoices THEN Np=1
21566    UNTIL Use(Np)
21569    GOTO Change_params
21572  CASE -1,-162              ! CONTROL-CONTINUE or UP-ARROW keys pressed
21575    P=Np
21578    REPEAT
21581      Np=Np-1
21584      IF Np<1 THEN Np=Nchoices  ! wraparound
21587    UNTIL Use(Np)
21590    GOTO Change_params
21593  CASE -211 ! DEL CHR
21596    IF J>-1 AND J<LEN(TRIM$(A$)) THEN
21599      A$[J+1]=A$[J+2]&" "
21602      PRINT TABXY(42+J,Np+2);Ci$&A$[J+1,J+1]&Cn$&A$[J+2,36]
21605    END IF
21608    GOTO Get_keystroke
21611  CASE -196,-194  ! Rt arrow or Lft arrow pressed- move cursor indicating position in response-string
21614    IF J=0 THEN 21467
21617    J=J+195+K
21620    IF J<0 THEN J=0
21623    IF J>34 THEN J=34
```

```
21626   PRINT TABXY(42,Np+2);A$[1,J]&Ci$&A$[J+1,J+1]&Cn$&A$[J+2,36]    ! move the cursor
21629   GOTO Get_keystroke
21632 CASE -221    ! CLR LN key pressed· clear the response area
21635   PRINT TABXY(42,Np+2);CHR$(127)&RPT$(" ",38)&Cn$
21638   GOTO 21467
21641 CASE -176 ! PAUSE key
21644   PAUSE
21647 CASE -177  ! dump-alpha key pressed
21650   DUMP ALPHA
21653   GOTO Get_keystroke
21656 CASE -171  ! CAPS LOCK key pressed
21659   Capslock=NOT Capslock
21662   CONTROL KBD:Capslock
21665   BEEP 300+300*Capslock,.1
21668   GOTO Get_keystroke
21671 CASE 8,15,31 ! CTL-H, CTL-/, CTL-?
21674   IF NPAR=8 THEN
21677     SELECT Helpsub
21680     CASE 1
21683       Manual_help(Np)
21686     CASE 2
21689       Fract_help(Np)
21692     CASE 3
21695       Magscan_help
21698     END SELECT
21701     Helped=1
21704     P=Np
21707     GOTO Begin
21710   ELSE
21713     CALL Clunk
21716     GOTO Get_keystroke
21719   END IF
21722 CASE <32,>126  ! invalid key
21725   Clunk
21728   GOTO Get_keystroke
21731 CASE ELSE
21734     !
21737   J=J+1  ! increment string·position
21740   IF J>50 THEN J=50
21743   A$[J,J]=CHR$(K)  ! define this character of the response-string
21746   IF J=1 THEN
21749     PRINT TABXY(41,Np+2);RPT$(" ",40)  ! clear response area
21752     A$=CHR$(K)&C$[1,49]
21755   END IF
21758   IF J<38 THEN PRINT TABXY(41+J,Np+2);A$[J,J]&Ci$&A$[J+1,J+1]&Cn$  ! print the new character in the reponse area, followed by
 a cursor
21761   GOTO Get_keystroke
21764 END SELECT
21767!
21770 Response$(Np)=TRIM$(A$)
21773 P=Np
21776 Numeric=0
21779 ON ERROR GOTO 21791
21782 Value=VAL(Response$(P))       ! if no error, string is a valid number
21785 OFF ERROR
21788 Numeric=1
21791 R1=Range(P,1)      ! minimum allowable numeric response
21794 R2=Range(P,2)      ! maximum   "         "       "
21797 IF R1=-1 AND R2=-1 THEN 21857     ! String response OK
21800 IF R1=-2 AND R2=-2 THEN           ! must be Y/N
```

```
21803   A=FNYes(Response$(P))
21806   IF A<>1 AND A<>0 THEN
21809     DISP FNH$("MUST ENTER")&FNB1$("Y")&FNH$("OR")&FNB1$("N")&FNH$("FOR THIS PARAMETER")
21812     GOTO Clunkout
21815   ELSE
21818     GOTO 21857
21821   END IF
21824 END IF
21827 IF NOT Numeric THEN
21830   DISP FNH$("YOU MUST ENTER A NUMERIC RESPONSE TO THIS PARAMETER")
21833   GOTO Clunkout
21836 END IF
21839 IF R1<>0 OR R2<>0 THEN
21842   IF Value<R1 OR Value>R2 THEN
21845     DISP FNH$(VAL$(Value)&" IS NOT WITHIN THE ACCEPTABLE RANGE ("&VAL$(R1)&" - "&VAL$(R2)&") "&"FOR THIS PARAMETER")
21848     GOTO Clunkout
21851   END IF
21854 END IF
21857 IF Exec THEN Done
21860 !
21863 !Special lines to handle spike-query input from first MANDAT form.
21866 IF NPAR=8 THEN
21869   IF Helpsub=2 AND P=3 AND Response$(P)="?" THEN
21872     Whichspike(0,0,0)     ! show user which spikes are defined
21875     Response$(P)="0"
21878     Helped=1
21881     P=Np
21884     GOTO Begin
21887   END IF
21890 END IF
21893 !
21896 PRINT TABXY(42,P+2);B$
21899 Np=P+1
21902 BEEP 660,.06
21905 GOTO Change_params
21908!
21911 Done:FOR I=1 TO Nchoices
21914   IF Response$(I)[1,2]="??" THEN Incomplete
21917 NEXT I
21920 OUTPUT KBD;Clear$;
21923 SUBEXIT
21926!
21929 Incomplete:! Check for undefined but essential parameters
21932 DISP FNH$("SORRY- YOU MUST ENTER A VALUE FOR EVERY ")&FNB1$("??")&FNH$("PARAMETER.")
21935 Clunk
21938 WAIT 2.5
21941 Np=I
21944 GOTO Change_params
21947 Retrn:RETURN
21950 Clunkout:Clunk
21953 WAIT 3
21956 Response$(P)="??"&"       "
21959 GOTO Inverse
21962 SUBEND! ------------------------------------------------------------
21965 !
21968 !
21971 Mbar:SUB Mbar(Mm$(*),INTEGER Barrel_position,Barrel_pos0,Min_barrel,Max_barrel,I_t)  ! manual barrel-rotation
21974 OUTPUT KBD;CHR$(255)&CHR$(75);
21977 PRINT TABXY(1,4);"USE KNOB OR ARROWS TO ROTATE BARREL."
21980 PRINT TABXY(1,6);"TO ROTATE BEYOND LIMITS OF FILAMENT-CONTACTS, PRESS THE "&FNH$("SHIFT")&" KEY ALSO."
```

```
21983  FOR I=0 TO 19
21986    ON KEY I LABEL "" CALL Clunk
21989  NEXT I
21992  ON KEY 9 LABEL "    ESCAPE" GOTO Exit
21995  A=KNOBX
21998  OUTPUT 8;Mm$(2+Mu,I_t)
22001  DISP "BARREL: ";Barrel_position
22004  ON KNOB .1 GOTO 22016
22007  ON KBD GOTO 22040
22010  GOTO 22010
22013 !
22016  STATUS 2,10;L
22019  Limit=(L()1)
22022  IF KNOBX>0 THEN
22025    GOSUB Increment
22028  ELSE
22031    GOSUB Decrement
22034  END IF
22037 !
22040  CALL Kybrd(KBD$,K)
22043  IF K=-176 THEN PAUSE
22046  Incr=(K=-185 OR K=-169 OR K=-162 OR K=-194)
22049  Decr=(K=-184 OR K=-172 OR K=-196 OR K=-170)
22052  IF Incr=0 AND Decr=0 THEN 22010
22055 !
22058  Limit=(K=-162 OR K=-170 OR K=-196 OR K=-194)
22061  IF Incr THEN GOSUB Increment
22064  IF Decr THEN GOSUB Decrement
22067  GOTO 22010
22070 !
22073 Increment:IF (Limit AND Barrel_position)=Max_barrel-3) OR (NOT Limit AND Barrel_position)2900) THEN
22076    BEEP 1000,.05
22079    GOTO 22097
22082  ELSE
22085    Barrel_position=Barrel_position+1
22088    CALL Brl(Barrel_position,-1,0,0)
22091    DISP "BARREL: ";Barrel_position
22094  END IF
22097  RETURN
22100 !
22103 Decrement:IF (Limit AND Barrel_position(=Min_barrel+3) OR (NOT Limit AND Barrel_position=0) THEN
22106    BEEP 100,.05
22109    GOTO 22127
22112  ELSE
22115    Barrel_position=Barrel_position-1
22118    CALL Brl(Barrel_position,-1,0,0)
22121    DISP "BARREL: ";Barrel_position
22124  END IF
22127  RETURN
22130  !
22133 Exit:OFF KBD
22136  OFF KNOB
22139  OUTPUT 8;Mm$(Mu,I_t)
22142  OUTPUT KBD;CHR$(255)&CHR$(75);
22145 SUBEND! ------------------------------------------------------------
22148!
22151!
22154 Filamentknob:SUB Filamentknob(Stripchart,Logmon,Time0,Collector$(*),INTEGER I_t,L,Isotope)
22157! manually change filament-currents, using knob or keyboard-keys
22160  OPTION BASE 1
```

```
22163  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
22166  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
22169  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
22172  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
22175  REAL Flag(2),Ok(4)
22178  INTEGER Pr
22181  OFF KEY
22184  T=Daly_ok(L)
22187  MAT Ok= (1)
22190  MAT Daly_ok= (0)
22193  Daly_ok(L)=T*Mu
22196  OUTPUT KBD;Clear$;
22199  A=KNOBX
22202  !
22205  ! put .2 amps through any fils with <.1a & check filament-flags
22208  FOR I=0 TO 1
22211    FOR J=1 TO 2
22214      IF Filament(I+2*J-1)<.1 THEN OUTPUT 8;Ff$(I,J),200
22217    NEXT J
22220  NEXT I
22223  CALL Flag(Flag(*),Mu,1)
22226  FOR I=1 TO 4 ! re-zero fil-currents if necessary
22229    IF Filament(I)<.1 THEN OUTPUT 8;Ff$(I)2,1+(I=2 OR I=4)),0
22232  NEXT I
22235  OUTPUT 8;Mm$(Mu,I_t)
22238  IF Flag(1)=2 OR Flag(1)=0 THEN Ok(1)=0
22241  IF Flag(1)<2 OR Nfils=1 THEN Ok(2)=0
22244  IF (Flag(2)<>1 AND Flag(2)<>3) THEN Ok(3)=0
22247  IF Flag(2)<2 THEN Ok(4)=0
22250  IF Filament(4))<.1 AND (Flag(2)<2) THEN Ok(4)=0
22253  !
22256  IF SUM(Ok)=0 THEN ! no filament contacts at all
22259    PRINT USING "18/,6X,K,6/";FNH$("CAN'T CHANGE FILAMENT CURRENT-- NO VALID FILAMENT CONTACTS")
22262    Clunk
22265    WAIT 2
22268    SUBEXIT
22271  END IF
22274  !
22277  IF NOT Stripchart THEN
22280    PRINT TABXY(14,10);"Use "&FNH$("KNOB")&" to change filament-currents."
22283    FOR I=1 TO 4
22286      IF Ok(I) THEN PRINT TABXY(14+(I-1)*17,14);FNH$("k"&VAL$(I))
22289    NEXT I
22292    IF Ok(1) THEN PRINT TABXY(10,15);"CENTER-SAMPLE"
22295    IF Ok(2) THEN PRINT TABXY(27,15);"SIDE-SAMPLE"
22298    IF Ok(3) THEN PRINT TABXY(44,15);"CENTER-PREHEAT"
22301    IF Ok(4) THEN PRINT TABXY(61,15);"SIDE-PREHEAT"
22304    GRAPHICS OFF
22307  END IF
22310  FOR I=0 TO 19
22313    ON KEY I CALL Clunk
22316  NEXT I
22319  IF SUM(Ok))1 AND Ok(1) THEN ON KEY 1 LABEL "CENTER-SAMPLE" GOSUB 22403
22322  IF Ok(2) THEN ON KEY 2 LABEL "SIDE-SAMPLE" GOSUB 22409
22325  IF NOT Stripchart THEN
22328    IF Ok(3) THEN ON KEY 3 LABEL "CENTER-PREHEAT" GOSUB 22415
22331    IF Ok(4) THEN ON KEY 4 LABEL "SIDE-PREHEAT" GOSUB 22421
22334  END IF
22337  ON KEY 9 LABEL "    ESCAPE" GOTO Exit
22340  GOSUB Dispcurrs
```

```
22343  I_t=2
22346  Enter_beam(Counts,Mv,L,2,Pr)
22349  WAIT .1
22352  ON CYCLE .195 GOSUB Query_beam
22355  ON KNOB .18 GOSUB Knob_change
22358  ON KBD GOSUB Keyboard
22361  GOTO 22361
22364   !
22367 Query_beam:Enter_beam(Counts,Mv,L,2,Pr)
22370  IF Pr>1 THEN Exit
22373  T=(TIMEDATE-Time0)/60
22376  IF Stripchart THEN
22379    IF NOT Logmon THEN DRAW T,Mv
22382    IF Logmon THEN DRAW T,LGT(ABS(Mv+(NOT Mv)))
22385  ELSE
22388    IMAGE 10X," mV",X,K," ",3X,K
22391    DISP USING 22388;Isotope,DROUND(Mv,4)
22394  END IF
22397  RETURN
22400 !
22403  Filnum=1
22406  GOTO Change_filnum
22409  Filnum=2
22412  GOTO Change_filnum
22415  Filnum=3
22418  GOTO Change_filnum
22421  Filnum=4
22424  !
22427 Change_filnum:BEEP 440,.1
22430 Dispcurrs:IF NOT Stripchart THEN
22433    FOR I=1 TO 4
22436      IF Ok(I) THEN
22439        U$=CHR$(128+(Filnum=I))&" "&VAL$(Filament(I))&" "&Cn$
22442        PRINT TABXY(12+0*(Filament(I)=0)+(I-1)*17,16);U$&RPT$(" ",10-LEN(U$))
22445      END IF
22448    NEXT I
22451  ELSE
22454    IMAGE " FILAMENT-KNOB ",5X,K," CENTER-FIL: ",X,D.3D,5X," ",K," ",5X,K,"SIDE-FIL: ",X,D.3D,4X,K
22457    DISP USING 22454;CHR$(128+3*(Filnum=1)),Filament(1),Isotope,CHR$(128+3*(Filnum=2)),Filament(2),Collector$(Mu)
22460  END IF
22463  Curr=Filament(Filnum)
22466  RETURN
22469                !
22472 Keyboard:CALL Kybrd(KBD$,K)
22475  SELECT K
22478  CASE -162,43              ! up-arrow or plus key pressed
22481    Change=.01
22484    GOTO Change_current
22487  CASE -170,45              ! down-arrow or minus key pressed
22490    Change=-.01
22493    GOTO Change_current
22496  CASE -176                 ! PAUSE key pressed
22499    PAUSE
22502  CASE ELSE
22505    Clunk
22508    RETURN
22511  END SELECT
22514  !
22517 Knob_change:Change=INT(1000.*KNOBX/2900)/1000
22520 Change_current:Curr=Filament(Filnum)+Change
```

```
22523  SELECT Curr
22526  CASE <0
22529    Curr=0
22532    BEEP 100,.05
22535  CASE >7
22538    Curr=7
22541    BEEP 2000,.05
22544  END SELECT
22547  Filament(Filnum)=Curr
22550  GOSUB Dispcurrs
22553  P=(Filnum>2)              ! preheat?
22556  OUTPUT 8 USING "4A,4Z";Ff$(P,1+(Filnum=2 OR Filnum=4)),FNF(Curr)
22559  RETURN
22562!
22565 Exit:OFF CYCLE
22568  OFF KNOB
22571  OFF KBD
22574  OUTPUT KBD;CHR$(255)&CHR$(75);
22577 SUBEND! ----------------    ----------------------------------------
22580!
22583!
22586 Autoform:SUB Auto_form(Run,Std,Escape,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),INTEGER Std_i
so(*))
22589  OPTION BASE 1
22592  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,0$,Clear$
22595  COM /Auto_form1/ Lastresponse$(*),Nresp(*),Last_runtype$,Normal(*),Response$(*),Prompt$(*)
22598  COM /Auto_form2/ Spike,INTEGER Iso(*),Niso,Rf,Magnet(*)
22601! Variables 1 thru 27 are the usual run/std variables, variables -3 thru 0 are the barrel#, run-type, isotopes, & sample name.
22604! All variables but -2 and 0 must be numeric.
22607! Variable -1 (isotopes) remains a string which must contain the isotopes, separated by commas.
22610!
22613 ! The Prompt$ array contains the prompts or labels of the parameters,
22616 ! the Response$ array as passed to the subprogram contains the default
22619 ! responses, and returns any changed responses.
22622 ! The Use array determines whether a given parameter can be accessed and
22625 ! changed.
22628 ! Parameters whose values must be freshly entered must be indicated with a
22631 ! passed response of "??".
22634 ! Move cursor with either KNOB, CONTINUE, or CONTROL-CONTINUE, or UP/DOWN arrows.
22637 ! Use right- or left-arrows to move cursor within a response, use the
22640 ! CLR LN key to blank-out a response.
22643 !
22646  OFF KEY
22649  Not_copied=1
22652  DIM A$[50],C(-3:0,2),Lr(-3:0),D(-3:0),Disp$[86],R$[110],Nu$(0:24)[6],S$[110],Isot(8),P$[50],Runtype$[8],Mcoef(3)
22655  INTEGER Co(0:1),Peak_inter,As,Use(-4:27),S_iso(3),Sref
22658  MAT Use= (1)
22661  Use(-4)=0
22664  FOR I=2 TO 4 ! don't access cf-only params if a single-fil run
22667    IF Response$(1)="1" THEN Use(I)=0
22670  NEXT I
22673 !
22676  DATA 47,1,14,2,47,2,14,3  ! columns, rows for response of params -3 to 0
22679  DATA 4,12,31,50           ! lengths of reponse-strings for params -3 to 0
22682  DATA 38,1,38,1            ! columns for prompts of params -3 to 0
22685!
22688  DATA RUN#,BARREL#,ELEMENT ....,ISOTOPES,SAMPLE NAME
22691  DATA STD-RUN#,"",ELEMENT,ISOTOPES,STD-RUN NAME
22694  DATA SINGLE(1) - TRIPLE(3),FOCUSING ISOTOPE (CF),CENTER-FIL BEAM (V),INITIAL CF (amps)
22697  DATA"DALY ENABLE (0,1,2)",CURRENT-1 (amps),RATE-1 (mA/SEC),WAIT-1 (min.),CURRENT-2 (amps),RATE-2 (mA/sec)
```

```
22700  DATA WAIT-2 (min.),DATA-WAIT (min.),ABORT CURRENT (amps)
22703  DATA MIN. BEAM (v),MAX. BEAM (v),DEFAULT CURR (amps),DEFAULT BEAM (v)
22706  DATA FIL. INCR/BLOCK (amps),MIN# BLOCKS,MAX# BLOCKS,MAX SIGMAMEAN(%), #SETS/BLOCK,MAX. GROWTH (%/minute),PREHEAT CF (amps),PR
EHEAT SF (amps),NORMSPIKE#
22709!
22712  READ C(*),Lr(*),D(*)
22715  IF Std THEN
22718    RESTORE 22691
22721    Lr(0)=10
22724  ELSE
22727    RESTORE 22688
22730  END IF
22733  FOR I=-4 TO 0
22736    READ Prompt$(I)
22739  NEXT I
22742  RESTORE 22694
22745  FOR I=1 TO 26
22748    READ Prompt$(I)
22751  NEXT I
22754!
22757  IF Lastresponse$(1)<>"" THEN
22760    MAT Response$= Lastresponse$
22763  ELSE
22766    DATA ??,??,??,??,??,??,??,??,??,"1",??,"5","0",??,"1","0","0",??,"1","9",??,??,"0","3","6",".05","15","100","0","0","0","",
""
22769    RESTORE 22766
22772    READ Response$(*)
22775  END IF
22778  IF NOT Std THEN
22781    Response$(-4)=VAL$(Run)
22784  ELSE
22787    Use(-3)=0
22790    Response$(-3)=""
22793  END IF
22796  Disp$="ENTER value (EXECUTE when done, k9 to escape. KNOB, (CTRL)CONTINUE moves cursor)"
22799  CONTROL 1,4;0 ! HOME key
22802  Capslock=1
22805  CONTROL KBD;Capslock ! CAPS LOCK on
22808 Begin:OUTPUT KBD;Clear$;
22811!
22814! print the prompts & responses for the run#, barrel#,run-type, isotopes, and sample-name
22817  PRINT TABXY(1,1);Prompt$(-4);TAB(LEN(Prompt$(-4))+2);FNH$(VAL$(Run));TAB(40);Prompt$(-3);TAB(50);Response$(-3)
22820  PRINT TABXY(1,2);Prompt$(-2);TAB(14);Response$(-2);TAB(38);Prompt$(-1);TAB(48);Response$(-1)
22823  PRINT TABXY(1,3);Prompt$(0);TAB(14);Response$(0)
22826   !
22829! print the prompts & reponses for the 26 run /std variables, in 2 columns
22832  FOR I=1 TO 13
22835    FOR J=0 TO 1
22838      IF Response$(1)<>"1" OR J OR (I<2 OR I>4) THEN
22841        Col=1+J*39
22844        Row=I+4
22847        Param=I+J*13
22850        PRINT TABXY(Col,Row);Prompt$(Param);TAB(24+Col);Response$(Param)
22853      END IF
22856    NEXT J
22859  NEXT I
22862  !
22865  IF NOT Std AND Not_copied=0 THEN
22868    P=0
22871    Np=0
```

```
22874  ELSE
22877    IF NOT Redo_screen THEN
22880      FOR P=-4+Std TO 26! must start on an allowed parameter
22883        IF Use(P) THEN 22889
22886      NEXT P
22889      Np=P
22892    END IF
22895  END IF
22898  Redo_screen=0
22901  !
22904! ------------------------------------------------------------------
22907 Change_params:! move cursor, dashes, & inverse-video to next parameter
22910  DISP Disp$
22913  PRINT TABXY(0,18);"(((  for help with a parameter, press CTRL-H(elp)  )))"
22916  IF P<1 THEN
22919    Last_resp_len=Lr(P)
22922    Last_promptcol=D(P)
22925    Last_respcol=C(P,1)
22928    Last_row=C(P,2)
22931  ELSE
22934    Last_resp_len=12
22937    Last_promptcol=1+39*(P)13)
22940    Last_respcol=24+39*(P)13)
22943    Last_row=P+4-13*(P)13)
22946  END IF
22949 !
22952  IF Np<1 THEN
22955    Resp_len=Lr(Np)
22958    Promptcol=D(Np)
22961    Nrcol=C(Np,1)
22964    Row=C(Np,2)
22967  ELSE
22970    Resp_len=12
22973    Promptcol=1+39*(Np)13)
22976    Nrcol=24+39*(Np)13)
22979    Row=Np+4-13*(Np)13)
22982  END IF
22985!
22988  IF Np=-2 THEN  ! run-type or std-run type; show detailed prompt
22991    PRINT TABXY(1,3);RPT$(" ",80)
22994    IF NOT Std THEN
22997      PRINT TABXY(1,3);"(e.g. "&FNH$("Sr")&". To use std runs, enclose std-run# in asterisks, e.g. "&FNH$("*12*")&")"
23000    ELSE
23003      PRINT TABXY(1,3);"(must match an element defined on the DATA disk)"
23006    END IF
23009  END IF
23012  IF Use(P) THEN
23015    PRINT TABXY(Last_respcol,Last_row);RPT$(" ",Last_resp_len+1)! blank out response area
23018    PRINT TABXY(Last_promptcol,Last_row);Prompt$(P);TAB(Last_respcol+1);Response$(P)! restore display format for previous param
eter (remove dash-pointer)
23021    IF P=-2 AND Np<>-2 THEN ! run-type parameter; restore NAME line
23024      PRINT TABXY(1,3);RPT$(" ",80)
23027      PRINT TABXY(1,3);Prompt$(0);TAB(14);Response$(0)
23030    END IF
23033  END IF
23036  L1=LEN(Prompt$(Np))
23039  IF Np>0 THEN PRINT TABXY(Promptcol+1+L1,Row);RPT$("-",22-L1) ! add dash-pointer
23042  PRINT TABXY(Nrcol,Row);FNBl$(Response$(Np)[1,Resp_len-1]) ! print new response to be changed in flashing inverse video
23045  !
23048 Start_response:J=0      ! counter for position in response-string
```

```
23051  A$=RPT$(" ",Resp_len)
23054 Get_keystroke:ON KBD ALL GOTO 23090
23057  ON KNOB .07 GOTO 23066
23060  GOTO 23060
23063     !
23066  IF KNOBX>0 THEN
23069    ON KNOB 1 GOSUB Retrn ! to disable knob while processing knob-rotation
23072    GOTO Movecursor_down
23075  ELSE
23078    ON KNOB 1 GOSUB Retrn ! ditto
23081    GOTO Movecursor_up
23084  END IF
23087     !
23090  CALL Kybrd(KBD$,K)
23093  ON KBD ALL GOSUB Retrn ! disable live keyboard while processing keystroke
23096!
23099  SELECT K
23102  CASE -199   ! k9 key pressed, so escape
23105    Escape=1
23108    OUTPUT KBD;Clear$;
23111    SUBEXIT
23114  CASE -187   ! ENTER key pressed
23117    Exec=0
23120    GOTO Check_response
23123  CASE -168   ! EXECUTE key pressed
23126    Exec=1
23129    GOTO Check_response
23132  CASE -189,-170    ! CONTINUE or DOWN-ARROW key pressed
23135 Movecursor_down:P=Np
23138    REPEAT
23141      Np=Np+1
23144      IF Np>26 THEN Np=-3
23147    UNTIL Use(Np) AND (Response$(1)<>"1" OR (Np<2) OR (Np)4))
23150    GOTO Change_params
23153  CASE -1,-162    ! CONTROL-CONTINUE or UP-ARROW keys pressed
23156 Movecursor_up:P=Np
23159    REPEAT
23162      Np=Np-1
23165      IF Np<-3 THEN Np=26
23168    UNTIL Use(Np) AND (Response$(1)<>"1" OR (Np<2) OR (Np)4))
23171    Go_on=1
23174    GOTO Change_params
23177  CASE -196,-194   ! Rt arrow or Lft arrow pressed- move cursor indicating position in response-string
23180    J=J+195+K
23183    IF J<0 THEN J=0
23186    IF J>Resp_len-1 THEN J=Resp_len-1
23189    PRINT TABXY(Nrcol+1,Row);A$[1,J]&Ci$&A$[J+1,J+1]&Cn$&A$[J+2,Resp_len]    ! move the cursor
23192    GOTO Get_keystroke
23195  CASE -221  ! CLR LN key pressed; clear response area
23198    PRINT TABXY(Nrcol,Row);CHR$(127)&RPT$(" ",Resp_len)&Cn$
23201    GOTO Start_response
23204  CASE -176
23207    PAUSE
23210  CASE -177  ! dump alpha key pressed
23213    DUMP ALPHA
23216  CASE -171  ! CAPS LOCK key pressed
23219    Capslock=NOT Capslock
23222    CONTROL KBD;Capslock
23225    BEEP 300+300*Capslock,.1
23228    GOTO Get_keystroke
```

```
23231  CASE 8,15,31 ! CTL-H, CTL-?, CTL-/
23234    CALL Autohelp(Np,Std)
23237    Redo_screen=1
23240    P=Np
23243    GOTO Begin
23246  CASE <32,>126   ! non-alphanumeric key, so ignore
23249    BEEP 200,.1
23252    GOTO Get_keystroke
23255  CASE ELSE ! valid alphanumeric keystroke
23258!
23261!
23264    IF J=0 AND TRIM$(A$)="" THEN PRINT TABXY(Nrcol,Row);RPT$(" ",Resp_len) ! clear response area
23267    J=J+1 ! process character to put in resonse-string &increment string-position
23270    IF J>Resp_len THEN J=Resp_len
23273    A$[J,J]=CHR$(K) ! define this character of the response-string
23276    PRINT TABXY(Nrcol+J,Row);A$[J,J] ! print the new character in the reponse area
23279    GOTO Get_keystroke
23282  END SELECT
23285!
23288 Check_response:R$=TRIM$(A$[1,Resp_len])
23291  IF LEN(A$) THEN Response$(Np)=TRIM$(A$[1,Resp_len])
23294  IF Np<>-2 AND Np<>0 AND Np<>26 AND (NOT Outgas OR Response$(Np)<>"***") THEN
23297    ON ERROR GOTO Non_numeric
23300    Nresp(Np)=VAL(Response$(Np))
23303    OFF ERROR
23306  END IF
23309   !
23312   !-----------------------------------------------------------------
23315  N=Nresp(Np)
23318  P$=Response$(Np)
23321  SELECT Np ! check values for valid ranges
23324  CASE -3
23327    IF N<1 OR N>16 THEN
23330      DISP Ci$;" THE BARREL# MUST BE IN THE RANGE OF 1 TO 16 ";Cn$
23333      GOTO Clunkout
23336    END IF
23339    IF TRIM$(UPC$(Response$(-2)))="OUTGAS" THEN Outgas=1 ! in case EXECUTE key pressed just for barrel# input
23342    IF Outgas AND (Response$(0)="??") OR Response$(0)="" THEN Response$(0)="***"
23345    IF Sample_name$(N)<>"" THEN
23348 !    IF NOT Outgas THEN Response$(0)=Sample_name$(N)
23351      Response$(0)=Sample_name$(N)
23354      PRINT TABXY(C(0,1),C(0,2));Response$(0);RPT$(" ",Lr(0)-LEN(A$))  ! print the sample name
23357    END IF
23360  CASE -2
23363    Response$(-2)=TRIM$(Response$(-2)[1,8])  ! trim to correct length
23366    R$=UPC$(Response$(-2))
23369    IF R$="" THEN
23372      DISP FNH$("YOU MUST ENTER AN ELEMENT-TYPE")
23375      GOTO Clunkout
23378    END IF
23381    Outgas=POS(R$,"OUTGAS") ! indicates outgassing only
23384    IF NOT Outgas AND Was_outgas THEN
23387      FOR I=-3 TO 26 ! change all "***" responses to ?? if not an outgas run
23390        IF Response$(I)="***" THEN
23393          Response$(I)="??"
23396          IF (I<2 OR I>4) AND Response$(1)<>"1" THEN
23399            Nresp(I)=1
23402            Use(I)=1
23405          END IF
23408        END IF
```

```
23411     NEXT I
23414     Redo_screen=1
23417     Np=-1
23420     Was_outgas=0
23423     GOTO Begin
23426   END IF
23429   Preheat=(R$[1,1]="P")*Outgas  ! outgas preheat filaments only
23432   Sample=(NOT Preheat) ! outgas sample filaments only
23435   IF Outgas THEN   ! use "***" as reponses to all irrelevant params
23438     DATA -3,-2,1,4,6,7,8,9,10,11
23441     FOR I=-3 TO 26
23444       RESTORE 23438
23447       FOR J=1 TO 10
23450         READ U
23453         IF I=0 THEN 23471
23456         IF U=I AND (U<>4 OR Nresp(1)=3) THEN 23471
23459       NEXT J
23462       Response$(I)="***"
23465       Nresp(I)=0+(I=5) ! don't zero if the Daly-enable variable
23468       Use(I)=0
23471     NEXT I
23474     Np=1
23477     Redo_screen=1
23480     Was_outgas=1
23483     GOTO Begin
23486   END IF
23489   !
23492   IF P$="*?*" THEN ! request display of defined std-run variables
23495     GOSUB Load_stdvars
23498     OUTPUT KBD;Clear$;
23501     PRINT USING "10A,3X,6A,5X,6A,9X,10A,26X,7A,/";"STD-RUN#","TYPE","NAME","ISOTOPES","#FILS"
23504     FOR I=1 TO 32
23507       IF Stdvar(I,1) THEN
23510         IF I=17 THEN
23513           DISP "PRESS "&FNH$("CONTINUE")&" TO SEE MORE STANDARD RUNS"
23516           PAUSE
23519         END IF
23522         PRINT TAB(3);I;TAB(12);UPC$(Stdtype$(I));TAB(21);Std_name$(I);
23525         FOR J=1 TO 8
23528           IF Std_iso(I,J) THEN PRINT TAB(33+5*(J-1));Std_iso(I,J);
23531         NEXT J
23534         PRINT TAB(69);Stdvar(I,1)
23537       END IF
23540     NEXT I
23543     DISP "PRESS "&FNH$("CONTINUE")&" TO RETURN TO RUN-VARIABLE SCREEN"
23546     PAUSE
23549     GOTO Begin
23552   END IF
23555   !
23558   IF NOT Std AND NOT Outgas THEN  ! test for std-run# input
23561     Stdcopy=0
23564     P1=POS(P$,"*")
23567     P2=POS(P$[P1+1],"*")+P1
23570     IF P1 AND P2 AND P2>P1 THEN Stdcopy=VAL(P$[P1+1,P2-1])
23573     IF Stdcopy THEN
23576       GOSUB Load_stdvars
23579       Response$(-2)=Stdtype$(Stdcopy) ! transfer std-run vars to menu
23582       FOR K=1 TO 26
23585         Response$(K)=VAL$(Stdvar(Stdcopy,K))
23588       NEXT K
```

```
23591       Response$(-1)=VAL$(Std_iso(Stdcopy,1))
23594       FOR K=2 TO 8
23597         IF Std_iso(Stdcopy,K) THEN Response$(-1)=Response$(-1)&","&VAL$(Std_iso(Stdcopy,K))
23600       NEXT K
23603       Parse((Response$(-1)),Isot(*),Diso)
23606       MAT Iso= Isot
23609       Not_copied=0
23612       IF Response$(1)="1" THEN
23615         FOR I=2 TO 4
23618           Use(I)=0
23621         NEXT I
23624       ELSE
23627         FOR I=2 TO 4
23630           Use(I)=1
23633         NEXT I
23636       END IF
23639       GOTO Begin
23642     ELSE
23645       Runtype$=R$
23648     END IF
23651   END IF
23654 CASE -1   ! isotopes input
23657   IF NOT Outgas THEN
23660     Parse((Response$(Np)),Isot(*),Diso)
23663     MAT Iso= Isot
23666     IF Diso<2 THEN
23669       DISP Ci$;" NEED AT LEAST TWO ISOTOPES!! ";Cn$
23672       GOTO Clunkout
23675     END IF
23678   END IF
23681 CASE 0  ! name
23684   IF Std AND NOT Outgas THEN Response$(0)=TRIM$(Response$(0)[1,10])
23687 CASE 1  ! #filaments- blank out or restore params 2-4 as required
23690   IF N<>1 AND N<>3 THEN
23693     DISP FNM$("SINGLE-FIL.=1    TRIPLE-FIL.=3")
23696     GOTO Clunkout
23699   END IF
23702   IF Exec THEN Done
23705   IF N=1 THEN
23708     BEEP 660,.06
23711     IF Use(2)=1 OR N=1 THEN
23714       FOR I=2 TO 4
23717         Use(I)=0
23720         Response$(I)="0"
23723         PRINT TABXY(1,I+4);RPT$(" ",39)
23726       NEXT I
23729       P=Np
23732       Np=5+(Use(5)=0)
23735       GOTO Change_params
23738     END IF
23741   ELSE
23744     IF Use(2)=0 THEN
23747       Outer=(POS(UPC$(Response$(-2)),"OUTGAS") AND N=3) ! indicates an outgas run for a triple-filament
23750       IF Outer THEN Np=3
23753       FOR I=2 TO 4
23756         IF I=4 OR Outer=0 THEN
23759           Use(I)=1
23762           PRINT TABXY(1,I+4);Prompt$(I);TAB(25);Response$(I)
23765         END IF
23768       NEXT I
```

```
23771      END IF
23774    END IF
23777  CASE 17  ! default beam
23780    IF NOT Outgas AND (ABS(N)<0 OR ABS(N)>10) THEN Out_of_range
23783  CASE 14,15     ! beamsizes in volts
23786    IF NOT Outgas AND (N<.0001 OR N>10) THEN Out_of_range
23789  CASE 3  ! CF-only beam
23792    IF NOT Outgas AND (N<0 OR N>10) THEN Out_of_range
23795  CASE 4,6,9,13,16,24,25  ! filament currents in amperes
23798    IF N<0 OR N>7 THEN Out_of_range
23801  CASE 5  ! daly-enable code
23804    IF N<0 OR N>2 THEN Out_of_range
23807  CASE 7,10  ! filament-current take-up rates in mA/second
23810    IF N<.01 OR N>1000 THEN Out_of_range
23813  CASE 8,11,12  ! waits, in minutes
23816    IF N<0 OR N>480 THEN Out_of_range
23819  CASE 18  ! fil.-current increase/block
23822    IF N>.1 THEN Out_of_range
23825  CASE 19,20  ! #blocks
23828    IF NOT Outgas AND (N<1 OR N>80) THEN Out_of_range
23831  CASE 22  ! #sets/block
23834    IF NOT Outgas AND (N<5 OR N>40) THEN Out_of_range
23837  CASE 26  ! spike number
23840    IF POS(Response$(26),"?") THEN  ! if a question-mark response, let
23843      Whichspike(0,0,0)     ! user see which spikes are defined
23846      Response$(26)="0"
23849      Redo_screen=1
23852      P=Np
23855      GOTO Begin
23858    END IF
23861    IF N<0 OR N>10 THEN Out_of_range
23864  END SELECT
23867!
23870  IF Exec THEN Done
23873  BEEP 660,.06
23876  GOTO Movecursor_down
23879!
23882! ------    ----------------------------------------------------
23885 Done:FOR P=-3 TO 26
23888    IF Response$(P)[1,2]="??" AND Use(P)=1 THEN Incomplete
23891    ON ERROR GOTO 23897
23894    Nresp(P)=VAL(Response$(P))
23897    OFF ERROR
23900  NEXT P
23903 Check_values:! check that the run-variables are internally consistent
23906  IF NOT Outgas AND Nresp(1)=3 AND Nresp(4)=0 THEN
23909    DISP Ci$&" IF YOU'RE RUNNING A TRIPLE, YOU MUST WANT "&Cb$&" SOME "&Ci$&"CENTER-FIL CURRENT ! "
23912    Clunk
23915    P0=4
23918    GOTO Curswitch
23921  END IF
23924  IF NOT Outgas AND (Nresp(15)/(Nresp(14)+1.E-9)-1)<.1 THEN
23927    DISP Ci$;" MAX. BEAM MUST BE AT LEAST 10% GREATER THAN MIN. BEAM ";Cn$
23930    P0=15
23933    GOTO Curswitch
23936  END IF
23939  IF Nresp(17)>Nresp(14) THEN
23942    DISP Ci$;" DEFAULT BEAM MUST NOT BE MORE THAN MIN. BEAM ";Cn$
23945    P0=17
23948    GOTO Curswitch
```

```
23951  END IF
23954  IF NOT Outgas AND Nresp(13)<=Nresp(9) THEN
23957    DISP Ci$;" ABORT-CURRENT MUST BE GREATER THAN CURRENT-2 ";Cn$
23960    PO=13
23963    GOTO Curswitch
23966  END IF
23969  IF NOT Outgas AND (Nresp(20)<Nresp(19)) THEN
23972    DISP CHR$(129);" MAX# BLOCKS MUST NOT BE LESS THAN MIN# BLOCKS ";Cn$
23975    PO=20
23978    GOTO Curswitch
23981  END IF
23984  IF NOT Outgas THEN
23987    Parse((Response$(-1)),Isot(*),Diso)
23990    MAT Iso= Isot
23993  END IF
23996  IF Std THEN 24260
23999   ! check that the specified isotopes are valid for the specified element
24002   ! and spike (run variables only- pass for std variables).
24005   !
24008  IF NOT Outgas AND (Last_runtype$="" OR ((UPC$(TRIM$(Response$(-2)[1,8]))<>UPC$(Last_runtype$)))) THEN  ! get element-data fro
m disk
24011    Runtype$=TRIM$(Response$(-2)[1,8])
24014    IF NOT Checked_els THEN  ! recheck existing elements on system-disk
24017      Check_elements(Element$(*),1)
24020      Checked_els=1
24023    END IF
24026    FOR I=1 TO 20
24029      IF TRIM$(UPC$(Element$(I)))=UPC$(Runtype$) THEN
24032        ON ERROR GOTO 24053
24035        ASSIGN @Path1 TO "TYPE:INTERNAL"
24038        ENTER @Path1,I;Element$(I),Niso,Mcoef(*),Magnet(*),Co(*),Peak_inter,As,Rf,Nu$(*),Normal(*)
24041        Last_runtype$=Runtype$
24044        GOTO 24071
24047      END IF
24050    NEXT I
24053    OFF ERROR
24056    CALL Bad_vdisk(0,0)
24059    PO=-2
24062    GOTO Curswitch
24065  END IF
24068    !
24071  IF NOT Outgas AND Nresp(1)=3 AND Nresp(3) THEN  ! if a triple-filament run with a defined CF-only beam
24074    FOR I=0 TO Niso ! is specified focussing-isotope valid?
24077      IF Nresp(2)=Magnet(I,1) THEN 24101
24080    NEXT I
24083    DISP Ci$;" ";Nresp(2);" ISN'T DEFINED FOR ELEMENT ";Runtype$;" ";Cn$
24086    PO=1
24089    GOTO Curswitch
24092  END IF
24095     !
24098  IF NOT Outgas THEN
24101    FOR J=1 TO Diso! check that each isotope is valid for specified element
24104      FOR I=1 TO Niso
24107        IF Iso(J)=Magnet(I,1) THEN 24122
24110      NEXT I
24113      DISP Ci$;" ISOTOPE ";Iso(J);" ISN'T DEFINED FOR ELEMENT ";Runtype$&" ";Cn$
24116      PO=-1
24119      GOTO Curswitch
24122    NEXT J
24125  !
```

```
24120   Spike=Nresp(26)
24131   IF Normal(1) AND NOT Spike THEN
24134     A=0
24137     B=0
24140     FOR J=1 TO Diso! check that isotopes include both the reference isotope and the normalizing isotope
24143       IF Iso(J)=Magnet(Rf,1) THEN A=1
24146       IF Iso(J)=Normal(1) THEN B=1
24149     NEXT J
24152     IF A=0 OR B=0 THEN
24155       DISP Ci$;" NEED BOTH";Magnet(Rf,1);"AND";Normal(1);"IN LIST OF ISOTOPES FOR ELEMENT "&Runtype$&" "&Cn$
24158       P0=-2
24161       GOTO Curswitch
24164     END IF
24167   END IF
24170!
24173   IF Spike THEN ! check for spike on both disks
24176     IF Lastspike$<>Spike$ OR Lastspike$="" THEN
24179       ON ERROR GOTO 24206
24182       FOR K=1 TO 2
24185         FOR I=1 TO 10
24188           ASSIGN @Path1 TO "SPIKE:INTERNAL,4,"&VAL$(2-K)
24191           ON ERROR GOTO 24203
24194           ENTER @Path1,Spike;S$,Sref,S_iso(*)
24197           OFF ERROR
24200           GOTO 24221
24203         NEXT I
24206       NEXT K
24209       DISP Ci$;" CAN'T FIND DATA FOR SPIKE ";Spike;" ON EITHER DISK. ";Cn$
24212       P0=26
24215       GOTO Curswitch
24218   !
24221       Lastspike$=Spike$
24224     END IF
24227   !
24230     CALL Spike_check(Spike,Iso(*),S_iso(*),Sref)
24233     IF Spike=0 THEN
24236       DISP Ci$;" SPIKE#";Nresp(26);" REQUIRES DATA FOR ISOTOPES ";Sref;S_iso(1);S_iso(2);" ";Cn$
24239       P0=-1
24242       GOTO Curswitch
24245     END IF
24248    END IF
24251  END IF
24254   !
24257   !
24260  MAT Lastresponse$= Response$
24263  OUTPUT KBD;Clear$;
24266  SUBEXIT
24269!
24272 Incomplete:! Check for undefined but essential parameters
24275  DISP Ci$;" SORRY- YOU MUST ENTER A VALUE FOR EVERY "&FNB1$("??")&FNH$("PARAMETER")
24278  Clunk
24281  WAIT 2.5
24284  P0=P
24287  P=Np
24290  Np=P0
24293  GOTO Change_params
24296!
24299 Out_of_range:Clunk
24302  DISP Ci$;" VALUE IS NOT WITHIN ACCEPTABLE RANGE FOR THIS PARAMETER ";Cn$
24305  WAIT 2.5
```

```
24308  GOTO Change_params
24311 Curswitch:P=Np
24314  Np=P0
24317 Clunkout:Clunk
24320  WAIT 2.5
24323  Response$(Np)="??"
24326  GOTO Change_params
24329 Non_numeric:OFF ERROR   ! non-numeric response
24332  DISP FNM$("MUST HAVE A NUMERIC VALUE FOR THIS PARAMETER")
24335  Response$(Np)="??"&"        "
24338  GOTO Clunkout
24341 Retrn:RETURN
24344!
24347  OFF ERROR
24350  CALL Bad_vdisk(1,0)
24353  GOTO Begin
24356  !
24359 Load_stdvars:  ! load std-run variables from disk if not already in memory
24362  IF Stdvar(1,1)=0 THEN        ! load std-runvars into memory
24365    ON ERROR GOTO 24374
24368    ASSIGN @Path1 TO "STDVAR:INTERNAL,4,1"        ! try left drive 1st
24371    GOTO 24380
24374    ON ERROR GOTO 24347
24377    ASSIGN @Path1 TO "STDVAR:INTERNAL"        ! then right drive
24380    ON ERROR GOTO 24347
24383    ENTER @Path1;Stdvar(*),Std_iso(*),Std_name$(*),Stdtype$(*)
24386    OFF ERROR
24389  END IF
24392  RETURN
24395 SUBEND! ------------------------------------------------------------
24398  !
24401  !
24404 Press_graph:SUB Press_graph(Mm$(*),Candump)
24407  OPTION BASE 1
24410  DIM P1(2),P2(2),T(2),Started(2),L(2),P9(2,2),Gauge$(2)[8]
24413  OFF KEY
24416  OFF KNOB
24419  OUTPUT KBD;CHR$(255)&CHR$(75);
24422  GCLEAR
24425  GRAPHICS OFF
24428  DATA 60,4,-9,1,.625
24431  READ Xtime,Yheight,Ymin,L(x)
24434  DIM Response$(10)[36]
24437  MAT Response$= ("")
24440  Response$(1)="START GRAPHICS PRESSURE-MONITOR"
24443  IF Candump THEN Response$(2)="ENABLE AUTO PRTR-DUMP OF GRAPHICS"
24446  Response$(3)="DISABLE  ''     ''         ''    "
24449  Response$(4)="DOUBLE X-AXIS TIME (now "&VAL$(Xtime)&" mins)"
24452  Response$(5)="HALVE    ''    ''   ''    ''    "
24455  Response$(6)="INCREASE MAX. PRESS. (NOW 1E-"&VAL$(9-Yheight)&")"
24458  Response$(7)="DECREASE ''      ''        "
24461  Response$(10)="RETURN TO BMC"
24464  CALL Keymenu(Response$(x))
24467  ON KEY 0 LABEL "   START" GOTO Start
24470  IF Candump THEN ON KEY 1 LABEL "ON AUTO-DUMP" GOSUB Autdump
24473  IF Candump THEN ON KEY 2 LABEL "OFF AUTO-DUMP" GOSUB Offdump
24476  ON KEY 3 LABEL "DOUBLE TIME" GOTO Double_x
24479  ON KEY 4 LABEL "HALVE TIME" GOTO Halve_x
24482  ON KEY 5 LABEL "INCREASE MAX-P" GOTO Incr_p
24485  ON KEY 6 LABEL "DECREASE MAX-P" GOTO Decr_p
```

```
24488  ON KEY 9 LABEL "9: ESCAPE" GOTO 24731
24491  GOTO 24491
24494!
24497 Autdump:Dgr=1
24500  DISP FNH$("AUTO PRINTER-DUMP")
24503  BEEP 440,.1
24506  RETURN
24509 Offdump:Dgr=0
24512  DISP FNH$("CANCEL AUTO PRINTER-DUMP")
24515  BEEP 220,.1
24518  RETURN
24521 Double_x:Xtime=2*Xtime
24524  BEEP 660,.1
24527  GOTO 24440
24530 Halve_x:IF Xtime>=2 THEN Xtime=INT(Xtime/2)
24533  BEEP 220,.1
24536  GOTO 24440
24539 Incr_p:Yheight=Yheight+1
24542  BEEP 660,.1
24545  GOTO 24440
24548 Decr_p:IF Yheight>1 THEN Yheight=Yheight-1
24551  BEEP 220,.1
24554  GOTO 24440
24557 Start:Ymax=Ymin+Yheight
24560  OFF KEY
24563  ON KEY 9 LABEL "    ESCAPE" GOTO 24731
24566  !
24569  Axes(0,100,8,100,0,Xtime,Ymin,Ymax,"MINUTES","PRESSURE (Mbar)",0,-1,1)
24572  U$=Mm$(Mu,1)[8]
24575  Gauge$(1)="$OMW00:"&U$
24578  Gauge$(2)="$OMW02:"&U$
24581  T0=TIMEDATE
24584  MAT T= (0)
24587  MAT P2= (0)
24590  MAT Started= (0)
24593  R=Yheight/90
24596  CSIZE 3
24599  FOR I=0 TO 1
24602    GOSUB Legend
24605  NEXT I
24608  OUTPUT 8;"$OMW04:"&U$
24611  WAIT 2
24614  FOR I=1 TO 2
24617    OUTPUT 8;"$IDV"
24620    ENTER 8;Zero    !          zero-value, integrated for 2 seconds
24623    Zero=Zero+Zero/2
24626  NEXT I
24629  LOOP
24632    IF P2(1)>Xtime THEN
24635      IF Dgr THEN DUMP GRAPHICS
24638      GOTO 24569
24641    END IF
24644    FOR Gauge=1 TO 2
24647      OUTPUT 8;Gauge$(Gauge)
24650      WAIT 1+(NOT Started(1))
24653      OUTPUT 8;"$IDV"
24656      ENTER 8;P    ! source  pressure-reading
24659      Pres=1.E-9
24662      ON ERROR GOTO 24704
24665      IF P<Zero THEN P=Zero
```

```
24668     IF Gauge=1 THEN P2(2)=(P-Zero-2950)/1020-8! log(source-pressure)
24671     IF Gauge=2 THEN P2(2)=7.105E-6*(P-Zero)-8.969! "  "  tube-  "
24674     P2(1)=(TIMEDATE-T0)/60! time in minutes
24677     P1(1)=P0(Gauge,1)
24680     P1(2)=P0(Gauge,2)
24683     IF Started(Gauge) THEN CALL Thickpen(P1(*),P2(*),R,L(Gauge))
24686     P0(Gauge,1)=P2(1)
24689     P0(Gauge,2)=P2(2)
24692     Started(Gauge)=1
24695   NEXT Gauge
24698 END LOOP
24701   !
24704 Legend:AREA COLOR L(I+1),L(I+1),L(I+1)
24707 MOVE .85*Xtime,Yheight*(.9-.08*I)+Ymin
24710 RECTANGLE .1*Xtime,R,FILL
24713 IMOVE .05*Xtime,R
24716 LORG 4
24719 IF NOT I THEN LABEL "SOURCE"
24722 IF I THEN LABEL "TUBE"
24725 RETURN
24728  !
24731 OUTPUT KBD;CHR$(255)&CHR$(75);
24734 SUBEND! ------------------------------------------------------------
24737  !
24740  !
24743 Bad_vdisk:SUB Bad_vdisk(Std,Write) ! Bad run- OR std-run variable disk-read/write
24746 DIM A$(1:3,0:1)[12]
24749 DATA READ,STORE,RUN,STANDARD-RUN,FROM,ON
24752 READ A$(*)
24755 DISP FNH$(" UNABLE TO "&A$(1,Write)&" "&A$(2,Std)&" VARIABLE FILE "&A$(3,Write)&" DISK ")
24758 Clunk
24761 WAIT 3
24764 SUBEND ! ------------------------------------------------------------
24767 !
24770 !
24773 Check_elements:SUB Check_elements(Element$(*),INTEGER Auto)
24776 MAT Element$= ("")
24779 ON ERROR GOTO 24806
24782 ASSIGN @Path1 TO "TYPE:INTERNAL"
24785 ON ERROR GOTO 24800
24788 FOR I=1 TO 30
24791   ENTER @Path1,I;Element$(I)
24794   IF NOT Auto AND Element$(I)<>"" THEN PRINT TAB(9);I;TAB(14);"---- ";Element$(I)
24797   Z=Z+1
24800 NEXT I
24803 IF Z=0 THEN
24806   OFF ERROR
24809   Clunk
24812   PRINT FNH$("CAN'T GET ELEMENT DATA ---");CHR$(10)
24815   PRINT FNH$("PRESS")&FNB1$("CONTINUE")&FNH$("WHEN THE SYSTEM DISK IS IN THE RIGHT-HAND DRIVE")
24818   PAUSE
24821   GOTO 24779
24824 END IF
24827 OFF ERROR
24830 SUBEND ! ------------------------------------------------------------
24833 !
24836 !
24839 Thickpen:SUB Thickpen(P1(*),P2(*),Thickness,Density)
24842 ! draw a line from point P1(x,y) to P2(x,y) with thickness defined in terms of Y-units & line-density from 0 to 1
24845 OPTION BASE 1
```

```
24848  DIM Pgram(4,2)
24851 !
24854  Pgram(1,1)=P1(1)
24857  Pgram(1,2)=P1(2)-Thickness/2
24860  Pgram(2,1)=P1(1)
24863  Pgram(2,2)=P1(2)+Thickness/2
24866  Pgram(3,1)=P2(1)
24869  Pgram(3,2)=P2(2)+Thickness/2
24872  Pgram(4,1)=P2(1)
24875  Pgram(4,2)=P2(2)-Thickness/2
24878 !
24881  AREA COLOR Density,Density,Density
24884  PEN 1
24887  MOVE P1(1),P1(2)
24890  ON ERROR GOTO 24896
24893  PLOT Pgram(*),FILL
24896 SUBEND ! ----------------------------------------------------------------
24899 !
24902 !
24905 Daly_enable:SUB Daly_enable(INTEGER Daly)
24908  COM /Keyboard/ Cn$,C1$,Cb$,Cu$,Q$,Clear$
24911  OUTPUT KBD;Clear$;
24914  GRAPHICS OFF
24917  OFF KNOB
24920  PRINT TABXY(1,3);"SELECT STATUS OF DALY-DETECTOR: "
24923  PRINT TABXY(1,6);FNUn$("ENABLE- DATA")
24926  PRINT TABXY(20,6);"Use when necessary for beam tune up "&FNUn$("and");TABXY(20,7);"for data-taking with small beams."
24929  PRINT TABXY(1,9);FNUn$("ENABLE- TUNEUP");TABXY(20,9);"Use when necessary for beam tune-up,"
24932  PRINT TABXY(20,10);"but "&FNUn$("not")&" for data-taking."
24935  PRINT TABXY(1,12);FNUn$("DISABLE DALY");TABXY(20,12);"Don't use Daly under any circumstances."
24938  PRINT TABXY(20,13);"(Use this ONLY if the Daly is malfunctioning)"
24941  OFF KEY
24944  ON KEY 0 LABEL "ENABLE- DATA" GOTO 24962
24947  ON KEY 1 LABEL "ENABLE- TUNEUP" GOTO 24968
24950  ON KEY 2 LABEL "   DISABLE" GOTO 24974
24953  ON KEY 9 LABEL "    ESCAPE" GOTO 24977
24956  GOTO 24956
24959     !
24962  Daly=1
24965  GOTO 24977
24968  Daly=2
24971  GOTO 24977
24974  Daly=0
24977  OUTPUT KBD;Clear$;
24980 SUBEND! ----------------------------------------------------------------
24983!
24986!
24989 Shiftlabel:SUB Shiftlabel(Stripchart,Sample_name$(*),INTEGER Sample,Run,Auto,Nfils) ! label the BMC-defined shifted-softkeys
24992  DIM Keys$(9)[50]
24995  DATA MAGNET,ION OPTICS,BARREL,COLLECTORS,ISOTOPE-RATIO DATA,SPIKES,"MASS-SPEC STATUS (pressure, HV, time, contacts...)",AUTOM
ATIC-RUNNING VARIABLES
24998  DATA "",CHANGING ANY FILAMENT CURRENT USING KNOB
25001  READ Keys$(*)
25004  IF Auto THEN Keys$(8)="REVERT TO MANUAL RUNNING"
25007  IF NOT Auto THEN Keys$(8)="START AUTOMATIC RUNNING"
25010  OUTPUT KBD;CHR$(255)&CHR$(75);
25013  GRAPHICS OFF
25016  PRINTER IS CRT
25019  PRINT TABXY(1,1);"USE THESE "&FNUn$("SHIFTED")&" SOFTKEYS (k0-k9) TO OBTAIN OTHER FUNCTIONS FOR:"
25022  FOR Key=0 TO 9
```

```
25025    PRINT TABXY(1,Key+3);FNH$("k"&VAL$(Key))&" ----- ";Keys$(Key)
25028  NEXT Key
25031  PRINT TABXY(1,14);"Use  KNOB  with CTRL-key to change center-fil current,"
25034  IF Nfils=2 THEN PRINT TABXY(1,15);"  '    ''    ''  CTRL-SHIFT keys to change side-fil current."
25037  PRINT TABXY(1,15+Nfils);"Barrel#  "&VAL$(Sample)&" , Run#  "&VAL$(Run)&" ";TAB(30);FNH$(Sample_name$(Sample))
25040  Stripchart=0
25043 SUBEND! ---- ------------------------------------------------------------
25046 !
25049 !
25052 Gain:DEF FNGain(Counts,Constant(*),INTEGER Daly) ! ##### use when Daly nonlinearity is calibrated
25055  ! corrects for collector-nonlinearity
25058  IF NOT Daly THEN RETURN Counts/100
25061  IF Daly THEN RETURN Counts*(Constant(1)+Counts*Constant(2))
25064 FNEND
25067 !
25070 !
25073 Locate:SUB Locate(Ok,Nuclide$(*),INTEGER Niso,Coarsebin(*)) ! find peaks in a given magnet-interval, find peak-to-peak & half-
peak intervals
25076   !
25079  OPTION BASE 1
25082  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
25085  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
25088  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
25091  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
25094  COM /Magnet/ McoefO(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),MagnetO(0:24,2),Coarse_in
25097  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
25100  DIM Peak(3),Isotope(24),Calibr_mag(24),Calibr_isot(24),Peak_mv(100)
25103  DIM Xa(3,24),Xb(3,3),Xc(3,3),Xd(3,24),Mcoef(3,1),Xarray(24,3),Yvector(24,1)
25106  DIM Peaksize(24),Tr(3,2),Nu$(24)[6],NuO$(24)[6]
25109  INTEGER Pr,Halfpeak_mag(0:1),Isonew(24),Magnet(0:24,2)
25112 M:IMAGE "$OFJ",4Z
25115  I_t=2
25118  MAT Magnet= (0)
25121 Definescan:OUTPUT KBD;Clear$;
25124  PRINT TABXY(1,4);"ENTER THE MAGNET COARSE-RANGE (0-10) AND THE MAGNET-INTERVAL (300-9700)"
25127  PRINT TABXY(1,5);"IN WHICH YOU EXPECT TO FIND PEAKS."
25130  INPUT Coarse,Minmag,Maxmag
25133  IF Coarse<0 OR Coarse>10 OR (Maxmag-Minmag)<300 OR Minmag<300 OR Maxmag>9700 THEN 25121
25136  OUTPUT 8;Mm$(Mu,I_t)
25139  M=Mu
25142  OUTPUT 8 USING "2(4A,4Z)";"$OFK";Coarsebin(Coarse),"$OFJ",Minmag
25145  OUTPUT KBD;Clear$;
25148  IF Daly THEN
25151    PRINT "PLEASE CHOOSE THE COLLECTOR FOR THE PEAK-SEARCH SCAN..."
25154    ON KEY 0 LABEL "   CUP" GOTO 25163
25157    ON KEY 1 LABEL "  DALY" GOTO 25169
25160    GOTO 25160
25163    Mu=0
25166    GOTO 25181
25169    Mu=1
25172  ELSE
25175    Mu=0
25178  END IF
25181  OFF KEY
25184  Trigger_mv=.1*Mu+(2-Daly)*(NOT Mu)  ! peak-height indicating a valid peak
25187  DATA 0,0,0,0,0,0
25190  RESTORE 25187
25193  READ On_pk,Cal_num,Ok,Mip,Onpk_num,Max_pksize
25196  Ymin=-1-Mu  ! min. Y is .1mV for cup, .01mV for daly
25199  Ymax=1-2*Mu  ! max. Y is 10v for cup, 100mV for daly
```

```
25202  OUTPUT KBD;Clear$;
25205  MinmagO=Minmag
25208  MaxmagO=Maxmag
25211  Axes(0,100,28,100,Minmag,Maxmag,Ymin,Ymax,"MAGNET","mV BEAM",0,Mu,1)
25214  OUTPUT KBD;Clear$;
25217  ALPHA ON
25220  OUTPUT 8;Mm$(Mu,2)
25223  WAIT 1+5*(Mu(M)
25226  FOR M=MinmagO TO MaxmagO STEP 3   ! scan for all peaks >Trigger_mv Mv
25229    Enter_beam(Y,Mv,1,I_t,Pr)
25232    IF Pr>1 THEN 25184
25235    PLOT M,LGT(ABS(Mv+(NOT Mv))),-1-(M=MinmagO)
25238    Peak(3)=Peak(2)
25241    Peak(2)=Peak(1)
25244    Peak(1)=Mv
25247    DISP "MAGNET: ";CHR$(128+On_pk);M;Cn$,DROUND(Mv,2);"mV"
25250     ! have last 3 mag-values given a beam?
25253    IF On_pk THEN
25256      D=Peak(3)<=Trigger_mv AND Peak(2)<=Trigger_mv AND Peak(1)<=Trigger_mv
25259    ELSE
25262      D=Peak(3)>Trigger_mv AND Peak(2)>Trigger_mv AND Peak(1)>Trigger_mv
25265    END IF
25268     ! If On_pk=0, D=1 when last 3 readings were >trigger.  If On_pk=1, D=1 when last 3 readings were <trigger.
25271  !
25274    IF On_pk THEN
25277      Onpk_num=1+Onpk_num
25280      Peak_mv(Onpk_num)=Peak(1)
25283    END IF
25286    IF D=0 THEN Nextmag
25289    !
25292    IF NOT On_pk THEN
25295      Startpk_mag=M
25298    ELSE
25301      Cal_num=1+Cal_num
25304      Calibr_mag(Cal_num)=(Startpk_mag+M)/2.9
25307      Peaksize(Cal_num)=SUM(Peak_mv)/Onpk_num
25310      IF Peaksize(Cal_num)>Max_pksize THEN  ! largest peak so far?
25313        Maxpk_num=Cal_num
25316        Max_pksize=Peaksize(Cal_num)
25319      END IF
25322    END IF
25325    !
25328    MAT Peak_mv= (0)
25331    Onpk_num=0
25334    On_pk=(NOT On_pk)
25337 Nextmag:OUTPUT 8 USING M;M+2
25340  NEXT M
25343  Whoop
25346  ON KEY 0 LABEL " SCAN OK" GOTO Identify_iso
25349  ON KEY 1 LABEL "REPEAT SCAN" GOTO Definescan
25352  GOTO 25352
25355  !
25358 Identify_iso:OFF KEY
25361  Xspred=Maxmag-Minmag
25364  Yspred=Ymax-Ymin
25367  W=Xspred/50 ! rectangle width
25370  H=Yspred/8!     "    height
25373  Tr(1,1)=0 ! triangle (arrow-point) array
25376  Tr(1,2)=0
25379  Tr(2,1)=.8*W
```

```
25382  Tr(2,2)=H/2
25385  Tr(3,1)=-1.6*W
25388  Tr(3,2)=0
25391  AREA PEN 1
25394  FOR I=1 TO Cal_num
25397    IF I>1 THEN  ! erase old arrow
25400      P=I-1
25403      AREA PEN -1
25406      GOSUB Draw_arrow
25409      AREA PEN 1
25412    END IF
25415    P=I
25418    GOSUB Draw_arrow
25421    BEEP 300,.1
25424    DISP "ENTER THE MASS & NUCLIDE OF THIS PEAK (e.g. "&CHR$(34)"88,Sr"&CHR$(34)" "&CHR$(34)"160,NdO"&CHR$(34)" "&CHR$(34)"87,Sr(Rb)"&CHR$(34)")";
25427    INPUT M,Nu$(I)[1,6]
25430    Calibr_isot(I)=DROUND(INT(M+.1),3)
25433  NEXT I
25436  GOSUB Peak_interval
25439  GOTO Halfpeak
25442  !
25445 Draw_arrow:Ystart=Ymin+1.15*(LGT(Peaksize(P)+(Peaksize(P)<0))-Ymin)     ! start of crotch of arrow
25448  IF (Ystart-Ymin)>.8*Yspred THEN Ystart=Ymin+.2*Yspred  ! if arrow too high
25451  MOVE Calibr_mag(P)-W/2,Ystart+Tr(2,2)   ! draw an arrow-pointer to the peak
25454  RECTANGLE W,H,FILL
25457  MOVE Calibr_mag(P),Ystart
25460  IPLOT Tr(*),FILL
25463  RETURN
25466 !
25469 Peak_interval:Peak_inter=0   ! calculate the average mag-interval for 1 a.m.u.
25472  AREA PEN  1
25475  P=Cal_num
25478  GOSUB Draw_arrow
25481  AREA PEN 1
25484  FOR I=2 TO Cal_num
25487    Peak_inter=Peak_inter+(Calibr_mag(I)-Calibr_mag(I-1))/(Calibr_isot(I)-Calibr_isot(I-1))
25490  NEXT I
25493  Peak_inter=Peak_inter/(Cal_num-1)
25496  RETURN
25499   !
25502 Halfpeak:Above=0        ! determine half-peak offset
25505  Mip=0
25508  CSIZE 2
25511  DISP
25514  OUTPUT 8 USING M;Calibr_mag(Maxpk_num)
25517  FOR K=1 TO 12
25520    Enter_beam(Y,Mv,1,I_t,Pr)
25523    IF Pr>1 THEN Halfpeak
25526    IF K>4 THEN Mip=Mip+Mv/8
25529  NEXT K
25532  LORG 5
25535  FOR I=0 TO Peak_inter
25538    Halfpeak_mag(Above)=Calibr_mag(Maxpk_num)-Peak_inter/2+I
25541    OUTPUT 8 USING M;Halfpeak_mag(Above)
25544    WAIT (((NOT I) OR (I=Peak_inter/2)))
25547    Enter_beam(Y,Mv,1,I_t,Pr)
25550    IF Pr>1 THEN 25448
25553    T=Mv-Mip/2
25556    Y=LGT(ABS(Mv+(NOT Mv)))
25559    MOVE Halfpeak_mag(Above),Y*(Y>Ymin)+Ymin*(Y<=Ymin)
```

```
25562   LABEL ":"
25565   DISP "FINDING HALF-PEAK LOCATIONS:   ";Halfpeak_mag(Above);TAB(40);DROUND(Mu,2)
25568   IF ((Above=0) AND (I<0)) OR ((Above=1) AND (I>0)) THEN 25583
25571   Above=1+Above
25574   BEEP
25577   IF Above=2 THEN 25586
25580   IF Above=1 THEN I=Peak_inter/2-1
25583 NEXT I
25586 DISP
25589 Aside=(Halfpeak_mag(1)-Halfpeak_mag(0))/2
25592 Count=0
25595 L=1
25598 MAT Daly_ok= (0)
25601 Magnet0(1,2)=Calibr_mag(Maxpk_num)
25604 Magnet0(1,1)=Calibr_isot(Maxpk_num)
25607 Coarsemag(1)=Coarsebin(Coarse)
25610 REPEAT ! do a more precise measurement of half-peak setting
25613   Aside0=Aside
25616   Center_peak(0,1)
25619   Enter_beam(Cts,Mvp,L,I_t,1,.2,0) !         peaktop mV
25622   OUTPUT 8 USING M;Magnet0(1,2)+Aside !       switch to above-side
25625   Enter_beam(Cts,Mva,L,I_t,1,.2,0) !         above mV
25628   OUTPUT 8 USING M;Magnet0(1,2)-Aside !       switch to below-side
25631   Enter_beam(Cts,Mvb,L,I_t,1,.2,0) !          below-mV
25634   R=(Mva+Mvb)/2/Mvp !                         ratio of half-pk to pktop mV
25637   Aside=Aside0-(R<.4)+(R>.6)
25640   Count=1+Count
25643 UNTIL Aside=Aside0 OR Count>12
25646 FOR I=1 TO Cal_num  ! center all isotopes found during scan
25649   Magnet0(1,2)=Calibr_mag(I)
25652   Magnet0(1,1)=Calibr_isot(I)
25655   Coarsemag(1)=Coarsebin(Coarse)
25658   M=Mu
25661   Mu=0
25664   OUTPUT 8;Mm$(0,2)
25667   IF M THEN WAIT 6
25670   Center_peak(0,(1))
25673   Calibr_mag(I)=Magnet0(1,2)
25676 NEXT I
25679 OUTPUT 8;Mm$(Mu,2)
25682 GOSUB Peak_interval
25685 DISP
25688  ! solve for either linear or least-squares quadratic curve to describe magnet-settings as a function of isotope
25691 IF Cal_num=2 THEN ! linear solution if only two peaks
25694   Mcoef(2,1)=(Calibr_mag(2)-Calibr_mag(1))/(Calibr_isot(2)-Calibr_isot(1))
25697   Mcoef(1,1)=Calibr_mag(1)-Mcoef(2,1)*Calibr_isot(1)
25700   Mcoef(3,1)=0
25703 ELSE! #peaks>=3
25706 ! solve for least-squares quadratic curve to describe magnet-settings as a function of isotope
25709   REDIM Xarray(Cal_num,3),Yvector(Cal_num,1)
25712   FOR J=1 TO Cal_num
25715     Xarray(J,1)=1
25718     Xarray(J,2)=Calibr_isot(J)
25721     Xarray(J,3)=Calibr_isot(J)^2
25724     Yvector(J,1)=Calibr_mag(J)
25727   NEXT J
25730   MAT Xa= TRN(Xarray)
25733   MAT Xb= Xa*Xarray
25736   MAT Xc= INV(Xb)
25739   MAT Xd= Xc*Xa
```

```
25742   MAT Mcoef= Xd*Yvector
25745  END IF
25748 !
25751  FOR I=1 TO Cal_num! check for improperly identified nuclides by checking residuals
25754    Residual=Mcoef(1,1)+Mcoef(2,1)*Calibr_isot(I)+Mcoef(3,1)*Calibr_isot(I)^2-Calibr_mag(I)
25757    IF ABS(Residual)>5 THEN
25760      DISP FNH$("AT LEAST 1 ("&VAL$(Calibr_isot(I))&"?) OF YOUR ISOTOPE IDENTIFICATIONS IS WRONG")
25763      Clunk
25766      WAIT 1
25769      GOTO Identify_iso
25772    END IF
25775  NEXT I
25778 !
25781 Add_iso: ! add isotopes that weren't found during scan
25784  OUTPUT KBD;Clear$;
25787  GRAPHICS OFF
25790  MAT Isonew= (0)
25793  PRINT USING "14/,K,/,K";"ISOTOPE, NUCLIDE OF PEAKS THAT "&FNUn$("WEREN'T")&" FOUND DURING SCAN?  (e.g. 85,Rb)","(6 characters
 max, no commas)"
25796  PRINT TABXY(1,18);"(press  CONTINUE  when done)"
25799  PRINT TABXY(1,1);
25802  FOR New_iso=0 TO 23-Cal_num
25805    I=0
25808    I1=New_iso+1
25811    DISP "NEW ISOTOPE, NUCLIDE #"&VAL$(I1)&" ";
25814    INPUT I,Nu0$(I1)
25817    IF I=0 THEN 25853
25820    M=Mcoef(1,1)+Mcoef(2,1)*I+Mcoef(3,1)*I^2
25823    IF M<300 OR M>9700 THEN
25826      DISP Ci$&" ISOTOPE ";I;" DOESN'T FALL WITHIN THE SPECIFIED COARSE-MAGNET RANGE "
25829      Clunk
25832      WAIT 3
25835      GOTO 25805
25838    ELSE
25841      Isonew(I1)=I
25844      PRINT "#"&VAL$(I1)&":  "&Nu0$(I1);I
25847    END IF
25850  NEXT New_iso
25853  FOR I=1 TO New_iso ! eliminate duplicate isotopes
25856    FOR J=1 TO Cal_num
25859      IF Isonew(I)=Calibr_isot(J) THEN Isonew(I)=0
25862    NEXT J
25865  NEXT I
25868  MAT Nuclide$= ("")
25871  Niso=0
25874  FOR I=1 TO 24 ! sort the new & scanned isotopes & place in Magnet array
25877    Miniso=1000
25880    Kmin=0
25883    FOR K=1 TO New_iso   ! find lowest new-isotope
25886      IF Isonew(K) AND Isonew(K)<Miniso THEN
25889        Miniso=Isonew(K)
25892        Nuclide$(I)=Nu0$(K)
25895        Kmin=K
25898      END IF
25901    NEXT K
25904    Jmin=0
25907    FOR J=1 TO Cal_num
25910      IF Calibr_isot(J) AND Calibr_isot(J)<Miniso THEN
25913        Miniso=Calibr_isot(J)
25916        Calibr_isot(J)=0
```

```
26096    PRINT USING "2/,K,/";FNH$("*** CAN'T GET SPIKE-DATA FROM DISK ***")
26099    Clunk
26102    WAIT 2
26105    GOTO 26033
26108  END IF
26111 SUBEND   ! ----------------------------------------------------------
26114 !
26117 !
26120 Sigpar:DEF FNSigpar(Sigma$)  ! Extract sigma-value from sigma-string- negative if in parens
26123  Paren=POS(Sigma$,"(")
26126  ON ERROR GOTO 26132
26129  RETURN SGN(.5-Paren)*VAL(Sigma$[Paren+1])
26132  RETURN 0
26135 FNEND ! ----------------------------------------------------------
26138!
26141!
26144 Magknob:SUB Magknob(INTEGER CoarseO,MagO,Coarsebin(*),On_daly) ! scan magnet with KNOB
26147  OPTION BASE 1
26150  INTEGER Mag
26153  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
26156  OUTPUT KBD;Clear$;
26159  IF On_daly THEN
26162    PRINT USING "18/,K,/,K,7/";"SORRY, YOU CAN'T DO KNOB-CONTROLLED SCANNING WHILE ON THE DALY.","USE THE SEMI-AUTO MAGNET-SCAN
 ROUTINE (k6) INSTEAD."
26165    Clunk
26168    WAIT 3
26171    GOTO Escape
26174  END IF
26177  Speed=100
26180  Coarse=CoarseO
26183  Mag=MagO
26186  GOSUB Display
26189  PRINT TABXY(1,8);"MAGNET-CONTROL KNOB ENABLED.  USE SOFTKEYS TO ESCAPE, TO CHANGE COARSE-RANGE,"
26192  PRINT TABXY(1,10);"OR TO CHANGE RESPONSE OF KNOB."
26195  OFF KEY
26198  ON KEY 9 LABEL "    ESCAPE" GOTO Escape
26201  ON KEY 0 LABEL "  COARSE UP" GOSUB Upcoarse
26204  ON KEY 1 LABEL " COARSE DOWN" GOSUB Downcoarse
26207  ON KEY 3 LABEL "   SPEED UP" GOSUB Speedup
26210  ON KEY 4 LABEL "  SLOW DOWN" GOSUB Slowdown
26213  A=KNOBX
26216  GOTO Knob
26219 !
26222 Upcoarse:Coarse=1+Coarse
26225  IF Coarse<11 THEN Mag=0
26228  IF Coarse>10 THEN Coarse=10
26231  OUTPUT 8 USING "4A,4Z";"$OFK",Coarsebin(Coarse)
26234  GOSUB Display
26237  BEEP 1000,.05
26240  RETURN
26243 Downcoarse:Coarse=Coarse-1
26246  IF Coarse>0 THEN Mag=9999
26249  IF Coarse<0 THEN Coarse=0
26252  OUTPUT 8 USING "4A,4Z";"$OFK",Coarsebin(Coarse)
26255  GOSUB Display
26258  BEEP 150,.05
26261  RETURN
26264 Speedup:Speed=2*Speed
26267  GOSUB Display
26270  BEEP 1500,.05
```

```
25919       Nuclide$(I)=Nu$(J)
25922       Jmin=J
25925     END IF
25928   NEXT J
25931   IF Kmin AND NOT Jmin THEN Isonew(Kmin)=0
25934   IF Miniso<>1000 THEN
25937     Magnet(I,1)=Miniso
25940     Niso=1+Niso
25943   END IF
25946 NEXT I
25949  !
25952 FOR I=1 TO Niso  ! define magnet values from quadratic curve
25955   Magnet(I,2)=Mcoef(3,1)*Magnet(I,1)^2+Mcoef(2,1)*Magnet(I,1)+Mcoef(1,1)
25958 NEXT I
25961 FOR I=1 TO 3
25964   Mcoef0(I)=Mcoef(I,1)
25967 NEXT I
25970 GRAPHICS OFF
25973 OUTPUT KBD;Clear$;
25976 MAT Magnet0= Magnet
25979 Ok=1
25982 SUBEND ! ----------
25985 !
25988 !
25991 Whichspike:SUB Whichspike(INTEGER Spike,Get_spike,Subflag)  !  get spike data from disk
25994 OPTION BASE 1
25997 COM /Spikedrun/ Spike$,Spikedrun_ratio$(*),Spike_ratio(*),Natural_ratio(*),INTEGER Idif(*),Spikedrun_iso(*),Spkdrun_ref
26000 OFF KEY
26003 OFF KNOB
26006 Subflag=0
26009 IF NOT Get_spike THEN CALL Dispikes
26012 FOR Drive=1 TO 0 STEP -1
26015   ON ERROR GOTO 26024
26018   ASSIGN @Path1 TO "SPIKE:INTERNAL,1,"&VAL$(Drive)
26021   GOTO 26030
26024 NEXT Drive
26027 GOTO 26087
26030 IF NOT Get_spike THEN
26033   Spike=0
26036   INPUT " WHICH SPIKE# DO YOU WANT TO LOOK AT? (press CONTINUE to escape)",Spike
26039   IF Spike=0 THEN SUBEXIT
26042 END IF
26045 ON ERROR GOTO 26087
26048 ENTER @Path1,Spike;Spike$,Spkdrun_ref,Spikedrun_iso(*),Spikedrun_ratio$(*),Idif(*),Spike_ratio(*),Natural_ratio(*)
26051 OFF ERROR
26054 OUTPUT KBD;CHR$(255)&CHR$(75);
26057 PRINT USING "4/,K,DD,2A,8X,K,K,2A,#";"SPIKE# ",Spike," ","NAME:  ",Spike$," "
26060 PRINT USING "10X,K,K,2A,3/,K,8X,K,8X,K,10/";"REFERENCE ISOTOPE:  ",Spkdrun_ref," ","RATIO ISOTOPES","SPIKE-RATIO","NAT
URAL-RATIO"
26063 FOR I=1 TO 3
26066   IF (Spikedrun_iso(I)<>0) THEN PRINT TABXY(3,9+I);Spikedrun_ratio$(I);TAB(23);Spike_ratio(I);
26069   IF I<3 THEN PRINT TABXY(43,9+I);Natural_ratio(I)
26072 NEXT I
26075 PRINT
26078 IF Get_spike THEN SUBEXIT
26081 GOTO 26033
26084  !
26087 Subflag=1
26090 OFF ERROR
26093 IF NOT Get_spike THEN
```

```
26273  RETURN
26276 Slowdown:Speed=Speed/2
26279  GOSUB Display
26282  BEEP 200,.05
26285  RETURN
26288!
26291 Display:DISP "COARSE-RANGE";Coarse;TAB(28);"MAGNET =";Mag;TAB(55);"SPEED =";INT(Speed)
26294  RETURN
26297!
26300 Knob:ON KNOB .05 GOTO Scan
26303  GOTO 26303
26306 Scan:S=KNOBX
26309  Mag=Mag+S*Speed/200
26312  IF Mag<0 THEN Mag=0
26315  IF Mag>9999 THEN Mag=9999
26318  OUTPUT 8 USING "1A,4Z";"$OFJ",Mag
26321  GOSUB Display
26324  IF Mag=0 THEN BEEP 100,.05
26327  IF Mag=9999 THEN BEEP 2500,.05
26330  GOTO 26303
26333!
26336 Escape:OUTPUT 8 USING "2(4A,4Z)";"$OFK",Coarsebin(Coarse0),"$OFJ",Mag0
26339  OUTPUT KBD;Clear$
26342 SUBEND!-----------------------------------------------------------
26345!
26348!
26351 Disp_elvals:SUB Disp_elvals(Runclass$,Nuclide$(*),Normal(*),Interfere(*),INTEGER Coarse,Ref,Hv0,Magnet(*),Niso,Inv)
26354  OPTION BASE 1
26357  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
26360  OUTPUT KBD;Clear$;
26363  FOR J=0 TO 24 STEP 8 ! display defined isotopes & magnet-values
26366    FOR I=J+1 TO 8+J
26369      IF I>Niso THEN 26384
26372      PRINT TRIM$(Nuclide$(I))&"-"&VAL$(Magnet(I,1));"  ";
26375    NEXT I
26378    PRINT
26381  NEXT J
26384  IF (Niso/8<>INT(Niso/8)) THEN PRINT
26387  CALL Magprt(Coarse,Magnet(*),Niso)
26390  PRINT USING "K,5X,K,K,8X,K,K";"ELEMENT-SERIES: "&FNM$(TRIM$(Runclass$)),"REF. PEAK: ",Ref,"DEFINED FOR HV=",Hv0
26393  PRINT "NORMALIZING RATIO:  ";
26396  IF Normal(1)=0 THEN PRINT "NONE"
26399  IF Normal(1) THEN PRINT VAL$(Ref)&"/"&VAL$(Normal(1))&"=",Normal(2)
26402  PRINT USING "K,1X,K,K,9X,K";Cu$&"MONITOR ISOT.","INTERFERING ISOT.","RATIO   "&Cn$
26405  FOR I=1 TO 4
26408    IF I=1 AND Interfere(I,1)=0 THEN PRINT TAB(4);"NONE";TAB(21);"NONE";TAB(45);"-----"
26411    IMAGE  4X,3D,15X,3D,15X,K,"/",K,"=",K
26414    IF Interfere(I,1) THEN PRINT USING 26411;Interfere(I,1),Interfere(I,2),Interfere(I,2),Interfere(I,1),Interfere(I,3)
26417  NEXT I
26420  PRINT
26423 SUBEND! -----------------------------------------------------------
26426!
26429!
26432 Superduperclunk:SUB Superduperclunk ! panic sound
26435  ON KEY 9 LABEL "   ESCAPE" GOTO 26456
26438  FOR I=1 TO 5
26441    Superclunk
26444    FOR J=1 TO 5
26447      Clunk
26450    NEXT J
```

```
26453  NEXT I
26456 SUBEND  ! ------------------------------------------------------------
26459 !
26462 !
26465 !convert:DEF FNClock_12$(Time$)  ! convert from 24-hour to 12 hour clock
26468  H=VAL(Time$) ! hour
26471  C1=POS(Time$,":")
26474  C2=POS(Time$[C1+C1],":")+C1
26477  Time$=Time$[1,C2-1]
26480  SELECT H
26483  CASE >12
26486    RETURN VAL$(H-12)&Time$[C1]&" P.M."
26489  CASE 12
26492    RETURN Time$&" P.M."
26495  CASE >0
26498    RETURN Time$&" A.M."
26501  CASE 0
26504    RETURN "12"&Time$[C1]&" A.M."
26507  CASE ELSE
26510    RETURN "???????"
26513  END SELECT
26516 FNEND !  ------------------------------------------------------------
26519 !
26522 !
26525 Isomag:DEF FNIsomag(Mcoef(*),Isotope)  ! return fine-magnet value for any isotope#
26528  RETURN Mcoef(1)+Isotope*(Mcoef(2)+Isotope*Mcoef(3))
26531 FNEND  ! ------------------------------------------------------------
26534  !
26537  !
26540 Newel:SUB Newel(Type,Interfere(*),Normal(*),Nuclide$(*),Element$(*),Runclass$,Early_exit,INTEGER Coarsebin(*),Niso,Ref,Rf,Inv,
Hv0)
26543  OPTION BASE 1
26546  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
26549  COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
26552  COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
26555  COM /Daly/ INTEGER Mu,Daly,Mm$(0:3,2)[8],Daly_ok(0:24),Ff$(0:1,2)[4]
26558  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
26561  COM /Filaments/ Filament(*),Fil$(*),INTEGER Nfils,Filnum
26564  DIM Runclass_$[10],Nuclide_$(0:24)[6],Mcoef_(3),Normal_(2),Interfere_(1,3)
26567  INTEGER Niso_,Magnet_(0:24,2),Coarsemag_(0:1),Peak_inter_,Aside_,Rf_,Inv_,Hv0_
26570  OFF KEY
26573  OFF KBD
26576  OUTPUT KBD;Clear$;
26579  Early_exit=0
26582  Type0=Type
26585  PRINT "PRESS "&FNH$("k0")&" TO CHANGE MAGNET-VALUES ONLY,";CHR$(10)
26588  PRINT "PRESS "&FNH$("k1")&" TO CHANGE RUNNING-DATA ONLY (normalization, interferences...),";CHR$(10)
26591  PRINT "PRESS "&FNH$("k2")&" TO CHANGE BOTH."
26594  PRINT USING "3/,K";"PRESS "&FNH$("k9")&" TO RETURN TO BMC"
26597  Magonly=0
26600  Datonly=0
26603  ON KEY 0 LABEL " MAGNET ONLY" GOTO 26618
26606  ON KEY 1 LABEL "RUN-DATA ONLY" GOTO 26624
26609  ON KEY 2 LABEL "     BOTH" GOTO 26627
26612  ON KEY 9 LABEL "    ESCAPE" GOTO Early_exit
26615  GOTO 26615
26618  Magonly=1
26621  GOTO 26627
26624  Datonly=1
26627  OUTPUT KBD;Clear$;
```

```
26630  OFF KEY
26633  ON KEY 9 LABEL "    ESCAPE" GOTO Early_exit
26636  IF NOT Datonly THEN
26639    CALL Locate(Ok,Nuclide$(*),Niso,Coarsebin(*))! peak-locating subprogram
26642    IF NOT Ok THEN Newel
26645    CALL Hv(Mm$(*),Hv,0,Mu,I_t)
26648    FOR I=1 TO 20
26651      IF UPC$(Element$(I)[1,2])="RE" THEN 26720
26654    NEXT I
26657    Magnet(0,1)=0
26660    Nuclide$(0)=""
26663    GOTO 26756
26666    !
26669    ASSIGN @Path1 TO "TYPE:INTERNAL"  ! get Re-element data from disk in right-drive
26672    ENTER @Path1,I;Runclass_$,Niso_,Mcoef_(*),Magnet_(*),Coarsemag_(*),Peak_inter_,Rside_,Rf_,Nuclide_$(*),Normal_(*),Inv_,Inte
rfere_(*),Hv0_
26675    IF ABS(Hv-Hv0_)<6 THEN
26678      FOR I=1 TO Niso_
26681        IF Magnet_(I,1)=187 THEN
26684          Magnet(0,2)=Magnet_(I,2)
26687          Magnet(0,1)=187
26690          Coarsemag(0)=Coarsemag_(1)
26693          Nuclide$(0)="Re"
26696          GOTO 26756
26699        END IF
26702      NEXT I
26705      Magnet(0,1)=0
26708      Magnet(0,2)=0
26711      Nuclide$(0)=""
26714    END IF
26717  END IF
26720  OFF KEY
26723  INPUT "ENTER REFERENCE ISOTOPE FOR AUTOMATIC RUNS ?   (0 to escape)",Ref
26726  IF Ref=0 THEN Early_exit
26729  FOR Rf=1 TO Niso
26732    IF Magnet(Rf,1)=Ref THEN 26750
26735  NEXT Rf
26738  PRINT USING "K,K,/";Ref," ISN'T DEFINED IN YOUR LIST OF ISOTOPES- PLEASE TRY AGAIN."
26741  Clunk
26744  GOTO 26720
26747  !
26750  L=Rf
26753  !
26756  IF NOT Magonly THEN
26759    MAT Normal= (0)
26762    Dispel(Element$(*))
26765    LOOP
26768      PRINT USING "K,/,K";"ENTER THE NUMBER (1-20) AND NAME (currently-defined elements shown above)","FOR THE NEW ELEMENT-TYPE
   (enter 0,0 to escape)"
26771      INPUT Type,Runclass$[1,10]
26774    EXIT IF Type)=0 AND Type(=20
26777      PRINT USING "/,10X,K,/";FNH$(" **** ELEMENT-TYPE NUMBER MUST BE (=20 **** ")
26780      Clunk
26783    END LOOP
26786    IF Type=0 THEN Early_exit
26789    Type0=Type
26792    OUTPUT KBD;Clear$;
26795    DISP "CALCULATE RATIOS AS ";VAL$(Ref)&"/"&"ISOTOPE (+1)  OR ISOTOPE/";VAL$(Ref);"  (-1)";
26798    INPUT Inv
26801    IF Inv=0 THEN Early_exit
```

```
26804   IF (Inv(>1) AND (Inv(>-1) THEN 26795
26807   MAT Normal= (0)
26810   PRINT TABXY(1,10);"WHAT ISOTOPE, IF ANY, DO YOU WANT TO USE TOGETHER WITH "&VAL$(Ref)&" FOR";TABXY(1,11);"FRACTIONATION NOR
MALIZATION?";
26813   PRINT TABXY(1,13);"(EXAMPLE: 88 FOR Sr";TABXY(1,15);"(press "&FNUn$("CONTINUE")&" if fractionation-normalization isn't poss
ible)"
26816   INPUT Normal(1)
26819   OUTPUT KBD;Clear$;
26822   IF Normal(1) THEN
26825     FOR I=1 TO Niso
26828       IF Normal(1)=Magnet(I,1) THEN 26846
26831     NEXT I
26834     Clunk
26837     DISP "*** ";Normal(1);" ISN'T IN YOUR LIST OF ISOTOPES ***"
26840     WAIT 3
26843     GOTO 26810
26846     DISP "ENTER THE STANDARD VALUE FOR THE NATURAL ";VAL$(Ref)&"/"&VAL$(Normal(1))&" RATIO";
26849     INPUT Normal(2)
26852   END IF
26855   Element$(Type)=Runclass$
26858   MAT Interfere= (0)
26861   PRINT USING "18/,K,/,K";"ENTER THE NUMBER (UP TO 4) OF ISOBARIC INTERFERENCES","FOR THE DATA-TAKING ISOTOPES:"
26864   INPUT Isobar_interf
26867   IF Isobar_interf=0 THEN 26945
26870   IF Isobar_interf)4 THEN 26861
26873   PRINT USING "18/,2(K,/)";"(The MONITOR isotope is the isotope whose peak-size is used to calculate the","peak-size of the I
NTERFERING isotope."
26876   PRINT USING "2(K,/)";"The INTERFERING isotope is an isotope whose mass coincides with the mass of the","isotope that you're
 really interested in."
26879   PRINT USING "K,/,K";"For example, in Strontium runs, Rb-85 is a MONITOR isotope for the INTERFERING","isotope Rb-87, which
interferes with Sr-87."
26882   FOR I=1 TO Isobar_interf
26885     DISP "MONITOR ISOTOPE #"&VAL$(I)&", INTERFERING ISOTOPE #"&VAL$(I)&" ?";
26888     INPUT Interfere(I,1),Interfere(I,2)
26891     A=0
26894     B=0
26897     FOR J=1 TO Niso
26900       IF Interfere(I,1)=Magnet(J,1) THEN A=1
26903       IF Interfere(I,2)=Magnet(J,1) THEN B=1
26906     NEXT J
26909     IF A=0 OR B=0 THEN
26912       DISP "**** BOTH ISOTOPES MUST EXIST IN THE LIST OF ISOTOPES FOR THIS ELEMENT ***"
26915       Clunk
26918       WAIT 3
26921       GOTO 26885
26924     END IF
26927     DISP "ENTER THE EXPECTED VALUE FOR THE "&VAL$(Interfere(I,2))&"/"&VAL$(Interfere(I,1))&" RATIO";
26930     INPUT Interfere(I,3)
26933   NEXT I
26936 ELSE
26939   Type=Type0
26942 END IF
26945 IF NOT Datonly THEN Hv0=Hv
26948 !
26951 ASSIGN @Path1 TO "TYPE:INTERNAL"
26954 OUTPUT @Path1,Type;Runclass$,Niso,Mcoef(*),Magnet(*),Coarsemag(*),Peak_inter,Aside,Rf,Nuclide$(*),Normal(*),Inv,Interfere(*),
Hv0
26957 ASSIGN @Path1 TO *
26960 SUBEXIT
26963 !
```

```
26966 Early_exit:OUTPUT KBD;Clear$;
26969  Early_exit=1
26972  PRINT USING "7/,K,/,K";"IN THE FOLLOWING DISPLAY, YOU MUST REQUEST A NEW ELEMENT TO RETURN TO THE BMC","WITHOUT PROBLEMS."
26975  PRINT USING "3/,18X,K";"Press  CONTINUE  when ready."
26978  PAUSE
26981 SUBEND! ----------------------------------------------------------------
26984 !
26987 !
26990 Dispel:SUB Dispel(Element$(*))  ! display all element-types defined on disk
26993  OUTPUT KBD;CHR$(255)&CHR$(75);
26996  GRAPHICS OFF
26999  PRINT TABXY(7,1);"NUMBER";TAB(20);"TYPE"
27002  ON ERROR GOTO 27032
27005  FOR I=1 TO 30
27008    IF Element$(I)()"" THEN
27011      Count=1+Count
27014      Side=(Count>15)
27017      IF Count=16 THEN PRINT TABXY(47,1);"NUMBER";TAB(59);"TYPE"
27020      Xpos=9+39*Side
27023      Ypos=Count+2 15*Side
27026      PRINT TABXY(Xpos,Ypos);I;TAB(4+Xpos);"--- ";Element$(I)
27029    END IF
27032  NEXT I
27035  OFF ERROR
27038  PRINT TABXY(1,18);
27041 SUBEND! ----------------------------------------------------------------
27044 !
27047 !
27050 H:DEF FNH$(Input$)  ! put a string in inverse-video
27053  RETURN CHR$(129)&" "&Input$&" "&CHR$(128)
27056 FNEND ! ----------------------------------------------------------------
27059 !
27062 !
27065 Un:DEF FNUn$(Input$,OPTIONAL P)  ! underline a string
27068  IF NPAR=2 THEN
27071    Printer=P
27074  ELSE
27077    Printer=0
27080  END IF
27083  IF Printer THEN RETURN CHR$(27)&"&dD"&Input$&CHR$(27)&"&d@"
27086  IF NOT Printer THEN RETURN CHR$(132)&Input$&CHR$(128)
27089 FNEND ! ----------------------------------------------------------------
27092 !
27095 !
27098 Bl:DEF FNBl$(Input$)  ! put a string in inverse, blinking video
27101  RETURN CHR$(131)&" "&Input$&" "&CHR$(128)
27104 FNEND ! ----------------------------------------------------------------
27107 !
27110 !
27113 Barnum:DEF FNBarnum(Barpos)  ! return the barrel# for a barrel-position
27116  IF Barpos=1 THEN RETURN 1   ! (the barrel# is the number stamped on the
27119  Barnum=22-5*Barpos          ! barrel; the barrel-position is the sequen-
27122  WHILE Barnum<1              ! tial position of a filament-block location
27125    Barnum=Barnum+16          ! in the barrel.
27128  END WHILE
27131  RETURN Barnum
27134 FNEND  ! ---------------------------------------------------------------
27137 !
27140 !
27143 Barpos:DEF FNBarpos(Barnum)   ! return barrel-position for a barrel#
```

```
27146  Barpos=Barnum*3-2
27149  WHILE Barpos>16
27152    Barpos=Barpos-16
27155  END WHILE
27158  RETURN Barpos
27161 FNEND ! ------------------------------ --------------------------------
27164 !
27167 !
27170 Fcal:DEF FNF(Analog_curr)  ! Change analog current in amps to the
27173 ! corresponding digital value necessary to to obtain that current
27176 ! 10/22/84 test indicates:  True current=0.88*(meter current)
27179 !                           (meter current)=1.12(digital output)+.21
27182 ! so (True Current)=(Digital Ouput)+.18
27185  RETURN MAX(INT(Analog_curr*1000-180),0)
27188 FNEND ! ------------------------------------------------------------------
27191 !
27194 !
27197 Dr:DEF FNDr(Number,Sig_figs)
27200  IF Number THEN
27203    M=10^(Sig_figs-INT(LGT(ABS(Number)))-1)
27206  ELSE
27209    M=1
27212  END IF
27215  RETURN INT(Number*M)/M
27218 FNEND ! ------------------------- ------------------------------------
27221 !
27224 !
27227 Wiggle:SUB Wiggle_barrel(INTEGER Barrel_position,Fil,I_t,REAL Flag(*),Mm$(*))
27230         ! wiggle the barrel a bit to try & regain contacts
27233  OUTPUT 8;Mm$(2,I_t)       ! enable barrel-motor
27236  CALL Brl(Barrel_position+6, 1, .6,0)   ! rotate barrel up 6 units, wait .6 sec
27239  CALL Brl(Barrel_position-8,-1, .8,0)   !   "      "    down 14 units, wait .8 sec
27242  CALL Brl(Barrel_position,Flag(Fil), .8) !   "      "    up 8 units to original
27245                              ! position, wait .8 sec, check contacts
27248 SUBEND  ! ------------------------------------------------------------------
27251 !
27254 !
27257 Mag_iso:DEF FNMag_iso(Mcoef(*),INTEGER Magnet)
27260   ! return isotope as function of fine-magnet setting
27263  IF Mcoef(3) THEN
27266    RETURN (-Mcoef(2)+SQR(Mcoef(2)^2-4*Mcoef(3)*(Mcoef(1)-Magnet)))/(2*Mcoef(3))
27269  ELSE
27272    RETURN (Magnet-Mcoef(1))/Mcoef(2)
27275  END IF
27278 FNEND
27281 !
27284 !
27287 Name:SUB Name(Sample_name$(*),INTEGER Estbar(*))
27290  ! If the barrel-contacts were tested, enter the run-names for all of the samples in the barrel
27293  COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,Q$,Clear$
27296  DIM C(16)
27299  OUTPUT KBD;Clear$;
27302  OFF KNOB
27305  PRINT USING "8/,K,2/,K";"This procedure will erase any previously-defined names.","Press  k0  to enter new names,  k9  to
 return to BMC."
27308  OFF KEY
27311  OFF KBD
27314  A=KNOBX
27317  ON KEY 0 LABEL "    START" GOTO 27329
27320  ON KEY 9 LABEL "    ESCAPE" GOTO 27383
```

```
27323  GOTO 27323
27326  !
27329  MAT Sample_name$= ("")
27332  OUTPUT KBD;Clear$;
27335  CONTROL KBD;1 ! caps lock ON
27338  FOR I=1 TO 16
27341    IF Statbar(I,1) THEN
27344      Count=1+Count
27347      C(I)=Count
27350      GOSUB Enter_name
27353    END IF
27356  NEXT I
27359 Edit:I=0
27362  PRINT TABXY(1,17);RPT$(" ",160)
27365  PRINT TABXY(1,17);"EDIT SAMPLE# ?  (press "&FNH$("CONTINUE")&" when done)       "
27368  INPUT I
27371  IF I>0 AND I<17 THEN
27374    GOSUB Enter_name
27377    GOTO Edit
27380  END IF
27383  SUBEXIT
27386 Enter_name:PRINT TABXY(1,17);"PLEASE ENTER THE NAME TO BE ASSIGNED TO THE RUN FOR BARREL# "&Ci$&" ";I;" "&Cn$&":"
27389  PRINT TABXY(1,18);"(press "&FNH$("CONTINUE")&" to escape)"
27392  LINPUT Sample_name$(I)[1,50]
27395  Sample_name$(I)=TRIM$(Sample_name$(I))
27398  IF Sample_name$(I)="" THEN Edit
27401  PRINT TABXY(1,C(I));RPT$(" ",80)
27404  IF Count=1 THEN PRINT TABXY(1,C(I));"BARREL#"
27407  PRINT TABXY(9,C(I));VAL$(I)&":";TAB(12);Sample_name$(I)
27410  RETURN
27413 SUBEND ! ------------------------------------ ----------------------
27416 !
27419 !
27422 Show_names:SUB Show_names(Sample_name$(*))
27425  OUTPUT KBD;CHR$(255)&CHR$(75);
27428  FOR I=1 TO 16
27431    IF Sample_name$(I)<>"" THEN
27434      Count=1+Count
27437      IF Count=1 THEN PRINT "BARREL#";
27440      PRINT TAB(9);I;TAB(15);Sample_name$(I)
27443    END IF
27446  NEXT I
27449  IF Count=0 THEN PRINT USING "2/,14X,K";"*** NO SAMPLE-NAMES HAVE BEEN DEFINED ***"
27452 SUBEND
27455 !
27458 !
27461 Autohelp:SUB Autohelp(Param,Standard) ! July 17, 1984, 12:56 pm
27464  OPTION BASE 1
27467  DIM S$(10)[160]
27470  SELECT Param
27473 !
27476  CASE -1
27479    RESTORE 27482
27482    DATA "$$$$$These are the isotopes that you wish to take isotope-ratio data for, excluding any isotopes used only for"
27485    DATA "isobaric interference monitoring.$The first isotope in the list must be the reference isotope.$$If the element that
"
27488    DATA "you choose for this run has a ratio defined for fractionation-normalization, then your list of isotopes must includ
e the"
27491    DATA "two isotopes used for the normalization.$$You may request up to 8 isotopes."
27494    N=4
```

```
 at a"
27659    DATA "rate of RATE-1 mA/second, followed by a wait of WAIT-1 minutes.$"
27662    DATA "Exceptions:  (1) the first 0.7 amps may assigned a RATE of 30 mA/second,  (2) if the filament was already preheated t
o"
27665    DATA "a given current, the assigned RATE will be doubled until that current is reached.$"
27668    DATA "The sample-filaments are then taken to a current of CURRENT-2 at a rate of RATE-2 mA/second,"
27671    DATA "followed again by a wait of WAIT-2 minutes.$"
27674    DATA "The beam is then centered and focused several times, followed by a wait of DATA-WAIT minutes just before the start
of data-taking."
27677    N=7
27680   !
27683  CASE 13
27686    RESTORE 27689
27689    DATA "$$$$$$$$$If ANALYST reaches the ABORT CURRENT on the sample-filament(s) [center if a single, sides if a triple] whi
le trying to"
27692    DATA "achieve either the MIN.-BEAM or the DEFAULT BEAM, the run is aborted and automatic-running sequence goes on to th
e next run."
27695    N=2
27698 !
27701  CASE 14,15,16,17! beam-size specs
27704    RESTORE 27707
27707    DATA "The beam-size of the MIP [most-intense peak] is kept at less than the MAX.-BEAM under all circumstances."
27710    DATA "As long as the sample-filament current (center-filament if a single-filament run, side-filaments if a triple-filament
 run)"
27713    DATA "is less than the DEFAULT CURRENT, the beam-size of the MIP will be maintained at no less than the MIN. BEAM"
27716    DATA "beamsize (in volts).$But if the sample-filament current exceeds the DEFAULT CURRENT, then the MIP beam will be"
27719    DATA "maintained at no less than the DEFAULT BEAM beamsize.$"
27722    DATA "The intent of the DEFAULT parameters is to allow you to specify both a desired beam-size (MIN.-BEAM to MAX.-BEAM),"
27725    DATA "and a 'not the best quality but I'll take it anyway' beamsize (DEFAULT-BEAM to MAX-BEAM)."
27728    DATA "$Tricky note:  To force ANALYST to try and get the MIN. BLOCKS at a beamsize of greater than the MIN. BEAM instea
d of just the"
27731    DATA"DEFAULT BEAM, enter the DEFAULT BEAM as its negative value."
27734    N=9
27737   !
27740  CASE 18  ! fil-incr/block
27743    RESTORE 27746
27746    DATA "$$$$$$$$After each block, the sample-filament current (center-fil if a single-fil run, side-fils if a triple-fil run)
 will be"
27749    DATA "increased by FIL. INCR./BLOCK amperes."
27752    N=2
27755   !
27758  CASE 19,20,21  ! min-max blocks, sigma
27761    RESTORE 27761
27764    DATA "ANALYST will take between MIN. #BLOCKS and MAX. #BLOCKS of data.$ANALYST will quit (go on to the next run)"
27767    DATA "as soon as it has taken MIN. #BLOCKS of data that have 1-sigmamean (%) precisions for each ratio of at least as goo
d"
27770    DATA "as the MAX. SIGMAMEAN(%) parameter.$"
27773    DATA "If the MAX. SIGMAMEAN(%) parameter is positive, then any ratio whose precision is within theoretical limits also
counts"
27776    DATA "as a 'good' ratio.  If MAX. SIGMAMEAN(%) is a negative value, then such ratios do not count towards the MIN. #
BLOCKS."
27779    N=5
27782  CASE 22
27785    RESTORE 27785
27788    DATA "$$$$$$ANALYST will take #SETS/BLOCK sets for each block of data-taking.$A SET is a step-scan across the isotopes
during data-taking."
27791    N=1
27794   !
27797  CASE 23
27800    RESTORE 27803
```

```
27497   !
27500  CASE -3
27503    IF Standard THEN SUBEXIT
27506    RESTORE 27509
27509    DATA "The BARREL # is the location in the barrel (1-16) of the sample to be run."
27512    N=1
27515     !
27518  CASE  2
27521    RESTORE 27524
27524    DATA "$The ELEMENT is the element for running, such as Pb, Sr, U, NdO...$It must be defined on the DATA disk and include
the isotopes that"
27527    DATA "you wish to take isotope-ratio data for.$To find out which elements are defined, press the CHANGE ELEMENT key during
the BMC."
27530    DATA "$To use a set of STD-RUN variables, enter the number of the STD-RUN enclosed in asterisks"
27533    DATA "(for example, *12*).  If you don't remember which STD-RUN you want, enter  *?*  for a summary of the"
27536    DATA "defined STD-RUN variables.$If you just want to outgas the sample, though, enter either OUTGAS (to outgas in the"
27539    DATA "running position)  or P-OUTGAS  (to outgas in preheat position)."
27542    N=0
27545     !
27548  CASE 0
27551    IF Standard THEN
27554      RESTORE 27557
27557      DATA "$$$$$$$$$The STD-RUN NAME is just the name (up to 10 characters) that you wish to assign to this standard-run."
27560    ELSE
27563      RESTORE 27566
27566      DATA "$$$$$$$$$The RUN NAME is just the name (up to 50 characters long) that will be assigned to this run."
27569    END IF
27572    N=1
27575   !
27578  CASE 1! single/triple
27581    RESTORE 27584
27584    DATA "$$$$$$$Enter 1 if the sample is a single-filament load, 3 if a triple (or  double)  filament load."
27587    N=1
27590 !
27593  CASE 2,3,4! foc-iso, CF-only beam, initial CF
27596    RESTORE 27599
27599    DATA "$$[FOR TRIPLE-FILAMENT RUNS ONLY]$$"
27602    DATA "The center filament is taken to an initial current of INITIAL CF AMPS, then the FOCUSING ISOTOPE (if nonzero) is
centered "
27605    DATA "and focused.$If nonzero, the FOCUSING ISOTOPE beam is then adjusted to the intensity"
27608    DATA "corresponding to the CENTER-FIL. BEAM by varying the center-filament current.$"
27611    DATA "If the FOCUSING ISOTOPE is 187 (Rhenium), its intensity will be maintained at the CENTER-FIL. BEAM throughout the
 run."
27614    DATA "$If the FOCUSING ISOTOPE is NOT Rhenium, then no attempt to control its beam-"
27617    DATA "size will be made once the side-filaments are turned on."
27620    N=7
27623 !
27626  CASE 5! Daly
27629    RESTORE 27632
27632    DATA "$$An entry of 0 indicates that the Daly is nonfunctional, and should not be used under any circumstances.$"
27635    DATA "An entry of 1 indicates that the Daly can be used for both beam tuneup and data-taking."
27638    DATA "Data will be taken with the Daly in all cases where all of the isotopes for data-taking are less than 35 mV in size.$
"
27641    DATA "An entry of 2 indicates that the Daly can be used for beam tuneup, but not under any circumstances for data-takin
g."
27644    N=4
27647  !
27650  CASE 6,7,8,9,10,11,12! current-rate-wait
27653    RESTORE 27656
27656    DATA "The sample filament(s) [center if a single-filament, sides if a triple-filament] are taken to a current of CURRENT-1
```

```
27803   DATA "$$$$$If the growth-rate of the beam exceeds MAX. GROWTH (%/MIN.) and the MIP (most- intense  peak) beam is more tha
n"
27806   DATA "the MIN. BEAM, the sample-filament current will be reduced by 2.5%.$If the MIP beam is less than the MIN. BEAM"
27809   DATA "value, ANALYST will tolerate up to twice the MIN. BEAM value.$Typical values are 2% - 3% per minute for"
27812   DATA "Sr and Nd.  A value of 100%/minute is equivalent to ignoring the beam growth."
27815   N=4
27818  !
27821 CASE 24,25
27824   RESTORE 27827
27827   DATA "$$$$$$$$Just after the running sample is rotated into position, the filaments of the preheat sample (at a barrel#
 of"
27830   DATA "one greater than the running sample) will be taken to currents of PREHEAT CF (A) amperes (center filament) and"
27833   DATA "PREHEAT SF (A) amperes (side filament)."
27836   N=3
27839  !
27842 CASE 26
27845   RESTORE 27848
27848   DATA "$$$$$$$$NORMSPIKE# is the number of the spike that the sample was spiked with  (possible only for elements with an
internal"
27851   DATA "ratio for fractionation- normalization, such as Sr, Nd, Sm...).$For a valid input, you must have already defined "
27854   DATA "a spike with this number.$Enter a value of 0 if the element is fractionation-normalizable, or if you haven't defined"
27857   DATA "such a spike.  If you enter a valid NORMSPIKE#, both the spike:sample ratio and the radiogenic-isotope ratio (if an
y), corrected for"
27860   DATA "fractionation and spike isotopes, will be calculated for each block."
27863   DATA "$To look at the spikes that are defined on the DATA disk, enter a question-mark (?) instead of a spike-number for t
he NORMSPIKE# value."
27866   N=6
27869 END SELECT
27872 REDIM S$(N)
27875 READ S$(*)
27878 Helpscreen(S$(*))
27881 SUBEND ! ----------------------------------------------------------
27884 !
27887 !
27890 Manual_help:SUB Manual_help(Param)
27893 OPTION BASE 1
27896 DIM S$(7)[160]
27899 SELECT Param
27902 CASE 1
27905   RESTORE 27908
27908   DATA "The SAMPLE NAME is the name that will be assigned to this run."
27911   N=1
27914 CASE 2
27917   RESTORE 27920
27920   DATA "$$$$$The ISOTOPES are the isotopes that you wish isotope-ratio data for.$The REFERENCE-ISOTOPE is the isotope that
will"
27923   DATA "appear in all of the ratios, and MUST be the first isotope that you enter.  Also, if the element you're running"
27926   DATA "has a ratio for internal fractionation-normalization, you MUST include the normalizing isotope (e.g. 88 for Sr)."
27929   DATA "You may request up to 8 isotopes.  Don't, however, include isotopes to be used for isobaric-interference monitoring"
27932   DATA "(such as Rb-85 for an Sr run)."
27935   N=5
27938 CASE 3
27941   RESTORE 27944
27944   DATA "$$$$$$$$A SET is a single sequence of step-scanning over the data-taking isotopes during a block."
27947   DATA "$You can specify from 1 to 10 SETS in a block."
27950   N=2
27953 CASE 4
27956   RESTORE 27959
27959   DATA "$$$$$$$$$A BLOCK is a complete cycle of data-taking, including peak-centering, backgrounds, and isobaric-interferen
ce monitoring."
```

```
27962   DATA "$You may specify from 1 to 80 blocks."
27965   N=2
27968 CASE 6,7,8
27971   RESTORE 27974
27974   DATA"$$$The BEAM WINDOW indicates the allowable beamsize for the Most Intense Peak (MIP).  If the MIP beamsize is found to
be"
27977   DATA "not within the specified BEAM WINDOW after a block of data, ANALYST will turn the filament current up or down until"
27980   DATA "the beamsize does fall within the BEAM WINDOW limits.  The BEAM WINDOW is defined by the MINIMUM BEAM and MAXIMUM BEA
M"
27983   DATA "values.  If  MINIMUM BEAM=0 and MAXIMUM BEAM=10, then no restrictions will be placed on the beam size."
27986   DATA"$ANALYST will not take the sample-filament current past the MAXIMUM FILAMENT- CURRENT however, regardless of the BEAM
WINDOW values."
27989   N=5
27992 CASE 9
27995   RESTORE 27998
27998   DATA "$$$$$If, after a block of data-taking, ANALYST determines that the rate of beam- growth exceeded the MAXIMUM BEAM-GRO
WTH"
28001   DATA "value, ANALYST will turn the sample- filament current down by about 2.4%.$A value of 100 is equivalent to disregardin
g"
28004   DATA "the rate of beam growth.$Typical useful values for Sr or Nd runs are 2% - 3% per minute."
28007   N=3
28010 CASE 11
28013   RESTORE 28016
28016   DATA "$$$$$$After the last block requested by the NUMBER OF BLOCKS value, ANALYST will turn  the sample and preheat filamen
t-currents"
28019   DATA "down to the values specified here.  The default values are simply the filament-currents in effect at this time, so"
28022   DATA "if you don't change the values, no change in the filament currents will occur when the data-blocks are done."
28025   DATA "The 4 values pertain to the Sample/Center-filament, Sample/Side-filaments, Preheat/Center-filament, and"
28028   DATA "Preheat/Side-filaments, respectively."
28031   N=5
28034 CASE 14
28037   RESTORE 28040
28040   DATA "$$$If the Daly-Enable Code is  0 , the Daly will be used for neither data-taking, nor beam tune-up under any"
28043   DATA "circumstances.  This code should be used  ONLY  if the Daly is malfunctioning.$A Daly-Enable Code of  1   allows"
28046   DATA "the Daly to be used both for beam-tuneup of small beams, and also for data-taking when all of the data-taking"
28049   DATA "isotopes are less than 35 millivolts.$A Daly-Enable Code of  2  allows the Daly to be used for beam tune-up"
28052   DATA "of small beams, but NOT, under any circumstances, for data-taking."
28055   N=5
28058 CASE 15  ! dump grafix after each block
28061   RESTORE 28064
28064   DATA "$$$$If you answer Y(es), the peak-jump graphics that appear during data-taking will be dumped to the printer at the
 end of each block.$"
28067   DATA "If you answer N(o), no such graphics-dump will occur."
28070   N=2
28073 END SELECT
28076 !
28079 REDIM S$(N)
28082 READ S$(*)
28085 Helpscreen(S$(*))
28088 SUBEND ! ------------------------------------------------------------
```

```
28091 !
28094 !
28097 Pattern:SUB Pattern  ! Hewlett-Packard "Pen" program from demo disk.
28100  INTEGER Polygon,Polygons,Side,Sides,Pen
28103  OFF KNOB
28106  Polygons=20
28109  Sides=3
28112  Pen=0
28115  ALLOCATE INTEGER X(0:Polygons-1,1:Sides),Y(0:Polygons-1,1:Sides)
28118  ALLOCATE INTEGER Dx(Sides),Dy(Sides)
28121  RANDOMIZE
28124  GINIT
28127  PLOTTER IS 3,"INTERNAL"
28130  GRAPHICS ON
28133  WINDOW 0,511,0,389
28136  PEN Pen
28139  ON KBD ALL GOTO Escape
28142  OFF KEY
28145  OFF KNOB
28148  A=KNOBX
28151  FOR Side=1 TO Sides
28154    X(0,Side)=RND*512
28157    Y(0,Side)=RND*390
28160    PLOT X(0,Side),Y(0,Side)
28163  NEXT Side
28166  IF Sides>2 THEN PLOT X(0,1),Y(0,1)
28169  GOSUB Define_deltas
28172  FOR Polygon=1 TO Polygons-1
28175    PENUP
28178    FOR Side=1 TO Sides
28181      Temp=X(Polygon-1,Side)+Dx(Side)
28184      IF Temp>511 THEN
28187        Dx(Side)=-Dx(Side)
28190      ELSE ! (it's not off right side)
28193        IF Temp<0 THEN Dx(Side)=-Dx(Side)
28196      END IF ! (off right side?)
28199      X(Polygon,Side)=X(Polygon-1,Side)+Dx(Side)
28202      Temp=Y(Polygon-1,Side)+Dy(Side)
28205      IF Temp>389 THEN
28208        Dy(Side)=-Dy(Side)
28211      ELSE ! (it's not off top)
28214        IF Temp<0 THEN Dy(Side)=-Dy(Side)
28217      END IF ! (off the top?)
28220      Y(Polygon,Side)=Y(Polygon-1,Side)+Dy(Side)
28223      PLOT X(Polygon,Side),Y(Polygon,Side)
28226    NEXT Side
28229    IF Sides>2 THEN PLOT X(Polygon,1),Y(Polygon,1)
28232  NEXT Polygon
28235  New=0
28238  ON CYCLE 10 GOSUB Define_deltas
28241  LOOP
28244    IF New=0 THEN
28247      Previous=Polygons-1
28250    ELSE
28253      Previous=(Previous+1) MOD Polygons
28256    END IF ! (new=0?)
28259    PENUP
28262    PEN -Pen
28265    DISABLE
28268    FOR Side=1 TO Sides
```

```
28271     PLOT X(New,Side),Y(New,Side)
28274   NEXT Side
28277   IF Sides>2 THEN PLOT X(New,1),Y(New,1)
28280   PENUP
28283   PEN Pen
28286   FOR Side=1 TO Sides
28289     Temp=X(Previous,Side)+Dx(Side)
28292     IF Temp>511 THEN
28295       Dx(Side)=-Dx(Side)
28298     ELSE
28301       IF Temp<0 THEN Dx(Side)=-Dx(Side)
28304     END IF
28307     X(New,Side)=X(Previous,Side)+Dx(Side)
28310     Temp=Y(Previous,Side)+Dy(Side)
28313     IF Temp>389 THEN
28316       Dy(Side)=-Dy(Side)
28319     ELSE
28322       IF Temp<0 THEN Dy(Side)=-Dy(Side)
28325     END IF
28328     Y(New,Side)=Y(Previous,Side)+Dy(Side)
28331     PLOT X(New,Side),Y(New,Side)
28334   NEXT Side
28337   IF Sides>2 THEN PLOT X(New,1),Y(New,1)
28340   ENABLE
28343   New=(New+1) MOD Polygons
28346 END LOOP
28349 Define_deltas: !
28352  FOR Side=1 TO Sides
28355    Dx(Side)=RND*3+2
28358    IF RND<.5 THEN Dx(Side)=-Dx(Side)
28361    Dy(Side)=RND*3+2
28364    IF RND<.5 THEN Dy(Side)=-Dy(Side)
28367  NEXT Side
28370  RETURN
28373 !
28376 !
28379 Escape:OFF KBD
28382  GRAPHICS OFF
28385 SUBEND
28388 !
28391 !
28394 Change_hv:SUB Change_hv  ! change some or all of the element HV values
28397  OPTION BASE 1
28400  DIM Mcoef(3),Normal(2),Interfere(4,3),Runclass$[10],Nuclide$(0:21)[6]
28403  INTEGER Niso,Magnet(0:24,2),Coarsemag(0:1),Peak_inter,Aside,Rf,Inv,Old_hv,New_hv
28406  !
28409  OUTPUT KBD;CHR$(255)&CHR$(75);
28412  OFF KEY
28415  OFF KNOB
28418  PRINT TABXY(1,6);"This subprogram will change the High-Voltage settings for all elements whose"
28421  PRINT USING "K,/";"present HV settings are within some specified range of a specified new value."
28424  PRINT "Please enter the NEW High-Voltage value and the RANGE from this value that"
28427  PRINT USING "K,2/";"the old HV values must fall within in order to be acceptable for the change."
28430  PRINT USING "K,20X,K";"Example:  8010,20      --  the HV of all elements whose present HV lies between","7990 and 8030 volts
will be changed to 8010."
28433  PRINT USING "/,K";"(enter 0,0 to escape to DMC)"
28436  !
28439  INPUT New_hv,Range
28442  IF New_hv=0 AND Range=0 THEN SUBEXIT
28445  IF New_hv>9999 OR Range>200 THEN
```

```
28448    Clunk
28451    DISP FNH$("*** THE NEW HV SHOULD BE <10,000 AND THE RANGE SHOULD BE <200 ***")
28454    WAIT 3
28457    GOTO 28439
28460  END IF
28463  !
28466  ON ERROR GOTO Bad_disk
28469  ASSIGN @Path1 TO "TYPE:INTERNAL"
28472  OFF ERROR
28475  OUTPUT KBD;CHR$(255)&CHR$(75);
28478  ON KEY 9 LABEL "    ESCAPE" GOTO Exit
28481  ON ERROR GOTO 28514
28484  DISP "Searching system disk..."
28487  FOR Type=1 TO 20
28490    ENTER @Path1,Type;Runclass$,Niso,Mcoef(*),Magnet(*),Coarsemag(*),Peak_inter,Aside,Rf,Nuclide$(*),Normal(*),Inv,Interfere(*)
,Old_hv
28493    Ocount=1+Ocount
28496    IF Old_hv<>New_hv AND ABS(New_hv-Old_hv)<=Range THEN
28499      OUTPUT @Path1,Type;Runclass$,Niso,Mcoef(*),Magnet(*),Coarsemag(*),Peak_inter,Aside,Rf,Nuclide$(*),Normal(*),Inv,Interfere
(*),New_hv
28502      PRINT "ELEMENT ";Runclass$;TAB(20);"HV Changed from ";Old_hv;"to";New_hv
28505    ELSE
28508      PRINT "ELEMENT ";Runclass$;TAB(20);"HV = ";Old_hv;"-- no change"
28511    END IF
28514  NEXT Type
28517  OFF ERROR
28520  !
28523  IF Ocount=0 THEN Bad_disk
28526 !
28529  Whoop
28532  DISP "press "&FNH$("CONTINUE")&" when ready to return to DMC..."
28535  PAUSE
28538  GOTO Exit
28541  !
28544 Bad_disk:Clunk
28547  PRINT USING "2/,10X,K";FNH$("*******  CAN'T READ SYSTEM DISK  *******")
28550  WAIT 4
28553  GOTO Exit
28556    !
28559 Exit:ASSIGN @Path1 TO *
28562 SUBEND
28565  !
28568  !
28571 Helpscreen:SUB Helpscreen(Input_string$(*),OPTIONAL In)
28574 ! prints out a message on the CRT with linefeeds at appropriate breaks
28577 ! between words.  Also inserts one linefeed for each $ symbol, and in-
28580 ! dents each line after a linefeed by Indent spaces.  Tricky, huh?
28583 !
28586  DIM Large_string$[1600],Line_string$[80],Rev_line$[80]
28589 !
28592  OUTPUT KBD;CHR$(255)&CHR$(75);
28595  PRINTER IS CRT
28598  IF NPAR=2 THEN Indent=In
28601  IF NPAR=1 THEN Indent=0
28604  Large_string$=""
28607  ! build a single large string out of the input-string array
28610  FOR I=1 TO SIZE(Input_string$,1)
28613    Large_string$=Large_string$&TRIM$(Input_string$(I))&" "
28616  NEXT I
28619  Maxlen=80
```

```
28622 !
28625  REPEAT
28628    LO=LEN(Large_string$)
28631    L=MIN(LO,Maxlen)
28634    Line_string$=TRIM$(Large_string$[1,L])
28637    Linefeed=POS(Line_string$,"$")  ! is there a linefeed-token?
28640    IF Linefeed THEN
28643      PRINT USING "K,/";RPT$(" ",80-Maxlen)&Line_string$[1,Linefeed-1]
28646      IF LO<=80 THEN Done
28649      Maxlen=80
28652      Large_string$=TRIM$(Large_string$[1+Linefeed])
28655    ELSE
28658      Rev_line$=REV$(Line_string$) ! find place for wordbreak
28661      Sr=POS(Rev_line$," ")
28664      IF LO>=Maxlen THEN
28667        S=L-Sr
28670      ELSE
28673        S=LO
28676      END IF
28679      PRINT RPT$(" ",80-Maxlen)&Line_string$[1,S]
28682      Large_string$=TRIM$(Large_string$[S+1])
28685      Maxlen=80-Indent
28688    END IF
28691  UNTIL L=0
28694 Done:DISP "press  CONTINUE  to go on,  DUMP ALPHA  for a hard copy"
28697  PAUSE
28700 SUBEND
28703 !
28706 !
28709 Fract_help:SUB Fract_help(Param)
28712 ! helpscreen for first manual-data request for
28715 ! a fractionation-normalizable element.
28718  OPTION BASE 1
28721  DIM S$(7)[160]
28724  !
28727  SELECT Param
28730  CASE 1   ! sample name
28733    RESTORE 28736
28736    DATA "$$$The SAMPLE NAME is just the name (up to 50 characters) that will be assigned to this run."
28739    N=1
28742  CASE 3   ! Spike Number
28745    RESTORE 28748
28748    DATA "$$$The SPIKE NUMBER will apply only if you have spiked your sample with a spike that has been defined on the DATA d
isk.$"
28751    DATA "If you have done so, then enter the number of that spike (the number that you assigned to it when you defined it on t
he disk)."
28754    DATA "Your isotope-ratio data will then be calculated corrected for both fractionation and spike isotopes (if there is
a radiogenic isotope"
28757    DATA "for this element), and the Spike:Sample ratio will also be calculated.$"
28760    DATA "If this is not such a spiked sample, though, just enter 0 as the spike number.$"
28763    DATA "To see which spikes are defined on the DATA disk, enter a Question-Mark instead of a spike-number.$"
28766    DATA "To define a new spike, exit to the BMC and then press SHIFT k5."
28769    N=7
28772  CASE 6   ! normalize ratios to 1st block?
28775    RESTORE 28778
28778    DATA "$$$If the isotope ratios of this sample will not have the natural ratios for the normalizing isotopes (for example,
 if the sample"
28781    DATA "is a spike, or if it's a sample spiked with an undefined spike), then you should answer Y(es)."
28784    DATA "The data will then use the ratio for the normalizing isotopes of the first block for fractionation-normalization of a
ll subsequent"
```

```
28787   DATA "Blocks.  You will then be able to do weighted averages on all the blocks from within ANALYST, without being bothered
by the"
28790   DATA "drift of fractionation.$Of course, the normalizing ratio will be wrong, but you can correct just the average rather
 than each"
28793   DATA "block for this bias.$For unspiked samples or for samples spiked with a spike that you have defined on the DATA disk (
SHIFT-k5 during the"
28796   DATA "BMC), though, you should answer N(o)."
28799   N=7
28802  END SELECT
28805  REDIM S$(N)
28808  READ S$(*)
28811  Helpscreen(S$(*))
28814 SUBEND
28817 !
28820 !
28823 Magscan_help:SUB Magscan_help
28826  OPTION BASE 1
28829  DIM S$(6)[160]
28832  DATA"$The magnet-scan starts at the mass-position of the Start Scan isotope, and goes up-mass at the Scan Speed to the E
nd Scan"
28835  DATA "isotope.  The scan is shown by real-time graphics, where the maximum beam that will fit on the graph is given by the M
ax. Beam on"
28838  DATA"Graph value.$The Y-axis of the graph can be either LINear or LOGarithmic.$"
28841  DATA "If you want to scan over a region that does not lie within the coarse-magnet range for the element currently in use,"
28844  DATA"enter a different coarse-magnet range (0-10) for the Coarse-Magnet Range.  The FORM screen will then ask you to enter
the"
28847  DATA"starting and ending positions of the scan in magnet units (0-9999) rather than mass units."
28850  READ S$(*)
28853  Helpscreen(S$(*))
28856 SUBEND
28859 !
28862 !
28865 Delta:SUB Delta(Delta$,Mean,Last_mean,Sigmean,Last_sigmean)
28868 ! determine if difference in ratios is significant & construct Delta$
28871  Percent_change=100*(Mean/Last_mean-1)
28874  Within_theor=(ABS(Percent_change)<2*SQR(Sigmean^2+Last_sigmean^2))
28877  Delta$[1,8]=TRIM$(VAL$(DROUND(Percent_change,2)))
28880  Delta$=TRIM$(Delta$)
28883  IF Percent_change>0 THEN Delta$="+"&Delta$[1,7]
28886  IF Within_theor THEN Delta$="("&Delta$&")"
28889 SUBEND
28892 !
28895 !
28898 Filtest:DEF FNFiltest(INTEGER Nfils,REAL Filflag)  ! valid-contact test?
28901  SELECT Nfils
28904  CASE 1
28907    RETURN Filflag=1 OR Filflag=3
28910  CASE >1
28913    RETURN Filflag=3
28916  CASE <1
28919    RETURN 0
28922  END SELECT
28925 FNEND
28928 !
28931 !
28934 DEF FNEven(N) ! is N even? return 1 if even, 0 if odd
28937  RETURN ABS((ABS(N)/2-INT(ABS(N/2)))<1.E-20
28940 FNEND
28943 !
28946 !
```