```
*
*                       PROGRAM PIBETA.F
*
*         Geant simulation code for the Pibeta experiment.
*         The active target used during the test of 1994,
*         the multi−wire proportional chambers, the plastic
*         veto staves, the calorimeter modules, the lead
*         shield and the cosmic veto counters are completely
*         defined and positioned here. This code has been used
*         for the design and understanding of the cluster
*         definitions and the triggers as discussed in chapter 7.
*         In addition, this code has been modified for the test
*         runs of 1993 and 1994. The simulation results which
*         are compared to the data in chapter 8 have been
*         obtained from the appropriate modifications to
*         this code. −−−
*
*                  By KETEVI ADIKLE ASSAMAGAN (KAA)
*
*CMZ :  3.13/05 16/06/89  16.58.03  by  Rene Brun
*CMZ :  3.13/05 16/06/89  16.58.03  by  Rene Brun
*CMZ :          02/03/89  08.37.38  by  Rene Brun
*CMZ :          02/03/89  08.37.38  by  Rene Brun
*CMZ :          02/03/89  08.37.38  by  Rene Brun
*
**********************************************************************
*                                                                    *
*             Test program for the CDHS set−up                         *
*       Authors     R.Brun, F.Carminati   *********                  *
*                                                                    *
**********************************************************************
*
      COMMON/PAWC/H(3000000)
      COMMON/GCBANK/Q(3000000)
      COMMON/QUEST/IQUEST(100)
*
*     −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
*
      CALL GZEBRA(3000000)
      CALL HLIMIT(−3000000)
*
*           Geant initialization.
      CALL UGINIT
*
*           Start events processing.
      CALL GRUN
*
*           End of run.
      CALL UGLAST
```

```
      STOP
      END

*     CMZ :          02/03/89  08.37.38  by  Rene Brun
*
*-- Author :    DEC-16-91 KETEVI A. ASSAMAGAN
*
*              REVISED AND ENLARGED : 06-13-1995 KAA
*
*
      SUBROUTINE UFILES
*
      OPEN(unit=4, FILE='pibeta.dat', STATUS='OLD')
      OPEN(unit=20,FILE='/fs22/users/ketevi/scratch/pion.dat',
     1     STATUS='OLD',FORM='UNFORMATTED')
*
      END
*CMZ :          13/05/89  19.04.22  by  Rene Brun
*
*-- Author :    DEC-16-91 KETEVI A. ASSAMAGAN
*
      SUBROUTINE UGINIT
*
************************************************************************
*                                                                      *
*             To initialise GEANT3 program and read data cards        *
*                                                                      *
************************************************************************
      COMMON/GCLINK/JDIGI,JDRAW,JHEAD,JHITS,JKINE,JMATE,JPART,
     1             JROTM,JRUNG,JSET,JSTAK,JGSTAT,JTMED,JTRACK,
     1             JVERTX,JVOLUM,JXYZ,JGARP,JGPAR2,JSKLT
*
      COMMON/GCOPTI/IOPTIM
      data IOPTIM/2/
      COMMON/GCUNIT/LIN,LOUT,NUNITS,LUNITS(5)
      INTEGER LIN,LOUT,NUNITS,LUNITS
      COMMON/GCMAIL/CHMAIL
      CHARACTER*132 CHMAIL
*
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     1     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
*
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,100),TOFD(100),IFLGK(100)
*
      COMMON/GCNUM/NMATE,NVOLUM,NROTM,NTMED,NTMULT,NTRACK,NPART,
     1            NSTMAX,NVERTX,NHEAD,NBIT
*
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
```

```
     1                  LVOLUM(15),LINDEX(15),INFROM,NLEVMX,
     1                  NLDEV(15),LINMX(15),GTRAN(3,15),GRMAT(10,15),
     1                  GONLY(15),GLX(3)
*
      COMMON/energies/calo1(240),calo2(240),calo3(240),tgt1(4),
     1                    tgt2(4),tgt3(4),stf(3),tpv1(20),tpv2(20),
     1                    tpv3(20),tcv1(5),tcv2(5),tcv3(5)
*
      COMMON/delali/pent(10),hexa(50),hexb(50),hexc(50),
     1                  hexd(40),hdh(20),veto(20),etarg,ewhole,
     1                  epveto,etgt,eclad,SQMA,W,etot,pwhole,
     1                  pes(240),pnu(240),pv(20),cv(5),wcx(2),
     1                  wcy(2),wcz(2)
*
      COMMON/MAWUSE/E1,X1,Y1,Z1,E2,X2,Y2,Z2,E3,X3,Y3,Z3
*
      COMMON/ketevi/thetal,phil,pgam(8),thetag(2),phig(2)
     1                  ,plpi0,elpi0,relang,beta,gamma
*
      COMMON/FLAG/IFLAG,IAD,jcount,vertex(3)
*
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     1       ,DEEMAX,EPSIL,STMIN,CFIELD,PREC,IUPD,ISTPAR,NUMOLD
*
*
*     --------------------------------------------------------------------
*
*             Open user files
*
      CALL UFILES
*
*             Initialize GEANT
*
      CALL GINIT
*
*             Prints version number
*
      WRITE(LOUT,1000)
*
*             Read data cards with FFREAD
*
      CALL GFFGO
*
*             Initialize GEANT/ZEBRA data structures
*
      CALL GZINIT
*
*             Initialize graphics package
*
```

```
*       CALL GDINIT
*
*              Geometry and materials description.
*
      CALL GEPIBETA
*
*              Particle table definition and energy loss initialization.
*
      CALL GPART
      CALL GPHYSI
*
*             Create a view bank
*
      CALL VIEWYZ(1)
      CALL UINIT
*
 1000 FORMAT(/,'  ****  GEANT PIBETA VERSION 3.21  **** : ',/)
      END
*CMZ :          12/06/89  17.04.08  by  Rene Brun
*
*-- Author :    DEC-16-91 KETEVI A. ASSAMAGAN
*
      SUBROUTINE UINIT
*
*********************************************************************
*                                                                   *
*              To book the user's histograms                        *
*                                                                   *
*********************************************************************
*
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(30),NAMEC(30)
     1 ,NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG ,STEP  ,SNEXT ,SFIELD
     1 ,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL,ISTOP ,IGAUTO,IEKBIN
     1 , ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN,NLVSAV,ISTORY
*
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     1      ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
*
      COMMON/energies/calo1(240),calo2(240),calo3(240),tgt1(4),
     1               tgt2(4),tgt3(4),stf(3),tpv1(20),tpv2(20),
     1               tpv3(20),tcv1(5),tcv2(5),tcv3(5)
*
      COMMON/delali/pent(10),hexa(50),hexb(50),hexc(50),
     1              hexd(40),hdh(20),veto(20),etarg,ewhole,
     1              epveto,etgt,eclad,SQMA,W,etot,pwhole,
     1              pes(240),pnu(240),pv(20),cv(5),wcx(2),
     1              wcy(2),wcz(2)
*
      COMMON/MAWUSE/E1,X1,Y1,Z1,E2,X2,Y2,Z2,E3,X3,Y3,Z3
```

```
*
      COMMON/ketevi/thetal,phil,pgam(8),thetag(2),phig(2)
     1             ,plpi0,elpi0,relang,beta,gamma
*
      COMMON/QUEST/IQUEST(100)
*
      CHARACTER*8 TAGS(287)
*
      data maxnst/100000/
      data thetal,phil,pgam,thetag,phig/14*0.0/
      data plpi0,elpi0,relang,beta,gamma/5*0.0/
      data epveto,etgt,eclad,etot,etarg,ewhole,pwhole/7*0.0/
      data igauto/1/
      data pent,hexa,hexb,hexc/160*0.0/
      data hexd,hdh,veto/80*0.0/
      data e1,x1,y1,z1,e2,x2,y2,z2,e3,x3,y3,z3/12*0.0/
      data pv,cv/25*0.0/
      data wcx,wcy,wcz/6*-100.0/
*
      DATA calo1,calo2,calo3/720*0.0/
      DATA tgt1,tgt2,tgt3,stf/15*0.0/
      DATA tpv1,tpv2,tpv3/60*0.0/
      DATA tcv1,tcv2,tcv3/15*0.0/
*
      DATA TAGS/'ETGT','ECLAD','EPVETO','ETARG',
     1          'P01','P02',
     1          'P03','P04','P05','P06',
     1          'P07','P08','P09','P10',
     1          'A01','A02','A03','A04','A05',
     1          'A06','A07','A08','A09','A10',
     1          'A11','A12','A13','A14','A15',
     1          'A16','A17','A18','A19','A20',
     1          'A21','A22','A23','A24','A25',
     1          'A26','A27','A28','A29','A30',
     1          'A31','A32','A33','A34','A35',
     1          'A36','A37','A38','A39','A40',
     1          'A41','A42','A43','A44','A45',
     1          'A46','A47','A48','A49','A50',
     1          'B01','B02','B03','B04','B05',
     1          'B06','B07','B08','B09','B10',
     1          'B11','B12','B13','B14','B15',
     1          'B16','B17','B18','B19','B20',
     1          'B21','B22','B23','B24','B25',
     1          'B26','B27','B28','B29','B30',
     1          'B31','B32','B33','B34','B35',
     1          'B36','B37','B38','B39','B40',
     1          'B41','B42','B43','B44','B45',
     1          'B46','B47','B48','B49','B50',
     1          'C01','C02','C03','C04','C05',
```

```
     1          'C06','C07','C08','C09','C10',
     1          'C11','C12','C13','C14','C15',
     1          'C16','C17','C18','C19','C20',
     1          'C21','C22','C23','C24','C25',
     1          'C26','C27','C28','C29','C30',
     1          'C31','C32','C33','C34','C35',
     1          'C36','C37','C38','C39','C40',
     1          'C41','C42','C43','C44','C45',
     1          'C46','C47','C48','C49','C50',
     1          'D01','D02','D03','D04','D05',
     1          'D06','D07','D08','D09','D10',
     1          'D11','D12','D13','D14','D15',
     1          'D16','D17','D18','D19','D20',
     1          'D21','D22','D23','D24','D25',
     1          'D26','D27','D28','D29','D30',
     1          'D31','D32','D33','D34','D35',
     1          'D36','D37','D38','D39','D40',
     1          'H01','H02','H03','H04','H05',
     1          'H06','H07','H08','H09','H10',
     1          'H11','H12','H13','H14','H15',
     1          'H16','H17','H18','H19','H20',
     1          'V01','V02',
     1          'V03','V04','V05','V06','V07',
     1          'V08','V09','V10','V11','V12',
     1          'V13','V14','V15','V16','V17',
     1          'V18','V19','V20',
     1          'ETOT','EWHOLE','W','RELANG','PTGT',
     1          'PCLAD','PPVETO','PTARG','PWHOLE','SQMA',
     1          'EG1','EG2','PV01','PV02','PV03','PV04',
     1          'PV05','PV06','PV07','PV08','PV09','PV10',
     1          'PV11','PV12','PV13','PV14','PV15','PV16',
     1          'PV17','PV18','PV19','PV20','CVFR','CVBA',
     1          'CVTO','CVSL','CVSR','MWPCX1','MWPCX2',
     1          'MWPCY1','MWPCY2','MWPCZ1','MWPCZ2'/
*
*     -----------------------------------------------------------------
*
*       Book histograms, manually (not digitized by routine HITS)
*
      IQUEST(10) = 65000
*
      CALL HROPEN(2,'PIBETA','pibeta.rz','NQ',8192,istat)
      CALL HBOOKN(9,'PIBETA',287,'PIBETA',2000,TAGS)
*
*      open (unit=17,file='module.dat',status='old')
*      do i=1,240
*         read (17,*) pes(i),pnu(1)
*         pnu(i) = pnu(i)/100
*      enddo
```

```
*
      END
*CMZ :          02/03/89  08.37.38  by  Rene Brun
*
*-- Author :    DEC-16-91 KETEVI A. ASSAMAGAN
*
      SUBROUTINE GEPIBETA
*
***********************************************************************
*                                                                    *
*             Routine to define the geometry of the set-up.           *
*                                                                    *
***********************************************************************
*
      COMMON/GCNUM/NMATE,NVOLUM,NROTM,NTMED,NTMULT,NTRACK,NPART,
     1             NSTMAX,NVERTX,NHEAD,NBIT
*
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(30),NAMEC(30)
     1 ,NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG ,STEP  ,SNEXT ,SFIELD
     1 ,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL,ISTOP ,IGAUTO,IEKBIN
     1 , ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN,NLVSAV,ISTORY
*
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     1      ,DEEMAX,EPSIL,STMIN,CFIELD,PREC,IUPD,ISTPAR,NUMOLD
      COMMON/GCOPTI/ IOPTIM
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     1             LVOLUM(15),LINDEX(15),INFROM,NLEVMX,
     1             NLDEV(15),LINMX(15),GTRAN(3,15),GRMAT(10,15),
     1             GONLY(15),GLX(3)
*
      DIMENSION PAR(3),PPAR(3),CAL(6),TUB(3)
      DIMENSION FIBER(3),CLAD(3),PLAS(11),CFC(3)
      DIMENSION X0(9)
      DIMENSION Y0(9)
      DIMENSION Z1(2),Z2(2),Z3(3)
      DIMENSION A1(2),A2(2),A3(3)
      DIMENSION WMAT1(2),WMAT2(2),WMAT3(3)
      DIMENSION AMYL(3),ZMYL(3),WMYL(3)
      DIMENSION WCI(3),WCO(3)
      DIMENSION AGAS(2),ZGAS(2),WGAS(2),WC1GAS(3),WC2GAS(3)
      DIMENSION AWIRE1(3),AWIRE2(3),CVT(3),CVF(3),CVB(3)
      DIMENSION CVS(3),ENCI(3),ENCO(3)
      DIMENSION TLEA(3),FLEA(3),BLEA(3),SLEA(3)
      DIMENSION WLSF(3),WLSB(3),WLST(3),WLSS(3)
      DIMENSION WCH2(2)
*
      CHARACTER*3 TAG(69)
      CHARACTER*4 ACR(69)
*
```

```
      INTEGER num(9),NMYL,NGAS
      INTEGER NLMAT1,NLMAT2,NLMAT3
*
      REAL DENS1,DENS2,DENS3,X,DMYL,DGAS
*
      DIMENSION PE1(11),PE2(11)
      REAL xp01,yp01,zp01,xp1,yp1,zp1,
     1     xp02,yp02,zp02,xp2,yp2,zp2,
     1     TP11,PP11,TP12,PP12,TP13,PP13,
     1     TP21,PP21,TP22,PP22,TP23,PP23
*
      DIMENSION HA1(11),HA2(11)
      REAL TA11,PA11,TA21,PA21,TA31,PA31,
     1     TA12,PA12,TA22,PA22,TA32,PA32
      REAL xa1,ya1,za1,xa2,ya2,za2
      REAL xa01,ya01,za01,xa02,ya02,za02
*
      DIMENSION HB1(11),HB2(11)
      REAL TB11,PB11,TB21,PB21,TB31,PB31,
     1     TB12,PB12,TB22,PB22,TB32,PB32
      REAL xb1,yb1,zb1,xb2,yb2,zb2
      REAL xb01,yb01,zb01,xb02,yb02,zb02
*
      DIMENSION HC1(11),HC2(11)
      REAL TC11,PC11,TC21,PC21,TC31,PC31,
     1     TC12,PC12,TC22,PC22,TC32,PC32
      REAL xc1,yc1,zc1,xc2,yc2,zc2
      REAL xc01,yc01,zc01,xc02,yc02,zc02
*
      DIMENSION HD1(11),HD2(11)
      REAL TD11,PD11,TD21,PD21,TD31,PD31,
     1     TD12,PD12,TD22,PD22,TD32,PD32
      REAL xd1,yd1,zd1,xd2,yd2,zd2
      REAL xd01,yd01,zd01,xd02,yd02,zd02
*
      DIMENSION HH1(11),HH2(11)
*
      DIMENSION VT1(11),VT2(11),VT3(11),VT4(11)
      REAL TV11,PV11,TV21,PV21,TV31,PV31,
     1     TV12,PV12,TV22,PV22,TV32,PV32
      REAL xv1,yv1,zv1,xv2,yv2,zv2
      REAL xv01,yv01,zv01,xv02,yv02,zv02
*
      REAL xv3,yv3,zv3,xv4,yv4,zv4
      REAL xv03,yv03,zv03,xv04,yv04,zv04
*
      DATA PAR/500.0,500.0,500.0/
      DATA PPAR/12.58,12.9,22.5/
      DATA TUB/0.0,1.905,3.5/
```

```
      DATA CAL/25.98,48.1,24.0,156.0,0.0,360.0/
      DATA PLAS/30.0,0.0,262,0.15875,2.04315,2.09475,9.0,0.15875,
     1          2.04315,2.09475,9.0/
*
      DATA CFC/12.9,13.0,30.0/
*
      DATA CVT/129.50,1.27,147.25/
      DATA CVF/131.59,123.70,1.27/
      DATA CVB/131.59,108.65,1.27/
      DATA CVS/1.27,112.4,142.25/
*
      DATA TLEA/132.04,2.5,138.44/
      DATA FLEA/132.04,121.2,2.5/
      DATA BLEA/132.04,103.70,2.5/
      DATA SLEA/2.5,103.70,132.17/
*
      DATA WLSF/132.04,1.27,1.27/
      DATA WLSB/132.04,1.27,1.27/
      DATA WLST/1.27,1.27,150.0/
      DATA WLSS/1.27,1.27,145.0/
*
      DATA Z1,Z2,A1,A2/6.0,1.0,55.0,53.0,12.01115,1.00797,
     1                 132.905,126.9044/
      DATA WMAT1,WMAT2/0.91549,0.08451,0.511548,0.488452/
*
      DATA FIBER,CLAD/0.138,0.138,3.5,0.150,0.150,3.5/
      DATA DENS1,DENS3,NLMAT1,NLMAT3/1.049,1.181,2,3/
      DATA DENS2,NLMAT2/4.526,2/
      DATA WMAT3/0.59985,0.08054,0.31961/
      DATA Z3/6.0,1.0,8.0/
      DATA A3/12.01115,1.00797,15.9994/
      DATA AMYL,ZMYL,WMYL,DMYL/12.01,1.01,16.,6.,1.,8.,5.,4.,2.,1.39/
      DATA NMYL/-3/
      DATA WCH2/1.0,2.0/
      DATA WCI/5.7405,6.2895,17.5/
      DATA WCO/11.733,12.287,27.0/
      DATA AGAS,ZGAS,WGAS/40.0,58.0,18.0,34.0,0.6765,0.3235/
      DATA DGAS,NGAS/1.85E-03,2/
      DATA WC1GAS/5.760,6.270,17.5/
      DATA WC2GAS/11.76,12.26,27.0/
      DATA AWIRE1,AWIRE2/0.0,10.0E-04,17.5,0.0,10.0E-04,27.0/
      DATA ENCI/4.501295,7.598705,5.0/
      DATA ENCO/10.30185,12.74815,5.0/
*
      DATA num/9,9,9,7,5,9,9,7,5/
      DATA X0/-1.264,-1.264,-1.264,-0.948,-0.632,
     1        -1.264,-1.264,-0.948,-0.632/
      DATA Y0/0.0,0.316,0.632,0.948,1.264,
     1        -0.316,-0.632,-0.948,-1.264/
```

```
*
      DATA TAG/'E01','E02','E03','E04','E05','E06','E07','E08','E09',
     1          'E11','E12','E13','E14','E15','E16','E17','E18','E19',
     1          'E21','E22','E23','E24','E25','E26','E27','238','E29',
     1          'E31','E32','E33','E34','E35','E36','E37',
     1          'E41','E42','E43','E44','E45',
     1          'S11','S12','S13','S14','S15','S16','S17','S18','S19',
     1          'S21','S22','S23','S24','S25','S26','S27','S28','S29',
     1          'S31','S32','S33','S34','S35','S36','S37',
     1          'S41','S42','S43','S44','S45'/
*
*      DATA PE/0.0,360.0,5,2,-11.0,0.0,2.4515,11.0,0.0,4.5344/
*      DATA xp0,yp0,zp0/0.0,0.0,37.0/
*
      DATA PE1/11.0,4.0,270.0,1.0469,0.0,2.8820,0.0,1.9364,
     1         0.0,5.3305,0.0/
      DATA PE2/11.0,2.0,270.0,1.6939,1.7812,2.8820,0.0,
     1         3.1332,3.2945,5.3305,0.0/
      DATA xp01,yp01,zp01,xp02,yp02,zp02/0.0,-2.8259,37.0,
     1                                   0.0,-1.0793,37.0/
*
      DATA HA1/11.0,3.240276,270.19166,1.4667,
     1         1.934324,3.712564,0.022746,2.712198,
     1         3.576776,6.864157,0.011140/
      DATA HA2/11.0,2.98973,269.812134,1.353115,
     1         1.784971,3.712564,0.021878,2.502126,
     1         3.299731,6.864157,0.011798/
      DATA xa01,ya01,za01,xa02,ya02,za02/0.0,-2.089452,37,
     1                                   0.0,-1.927617,37/
*
      DATA HB1/11.0,3.49,266.0,1.576722,1.845073,3.556092,
     1         354.34,2.915130,3.411386,6.574772,354.34/
      DATA HB2/11.0,3.49,274.0,1.576722,1.845073,3.556092,
     1         5.46,2.915130,3.411386,6.574772,5.46/
*
      DATA xb01,yb01,zb01,xb02,yb02,zb02/0.2226,-2.245926,37.0,
     1                                   -0.2226,-2.245926,37.0/
*
      DATA HC2/11.0,3.43,270.0,1.554285,1.936481,3.930567,0.0,
     1         2.873644,3.579709,7.266481,0.0/
      DATA HC1/11.0,3.54,270.0,1.6017715,1.995645,3.930567,0.0,
     1         2.961446,3.689086,7.266481,0.0/
      DATA xc02,yc02,zc02,xc01,yc01,zc01/0.0,-2.213965,37.0,
     1                                   0.0,-2.281609,37.0/
*
      DATA HD2/11.0,3.45,273.0,1.561733,1.882653,3.766625,
     1         359.40,2.8860385,3.480486,6.967328,359.40/
      DATA HD1/11.0,3.45,267.0,1.561733,1.882335,3.766625,
     1         0.60,2.8860385,3.480486,6.967238,0.60/
```

```
      DATA xd02,yd02,zd02,xd01,yd01,zd01/-0.01377,-2.224521,37.0,
     1                                  0.01377,-2.224521,37.0/
*
      DATA HH2/11.0,3.3426,273.398,1.55815,1.8827,3.7606,358.8018,
     1         2.88245,3.4805,6.9577,358.8028/
      DATA HH1/11.0,3.4522,266.602,1.55815,1.8827,3.7606,1.1982,
     1         2.88245,3.48045,6.9577,1.982/
*
      DATA VT3/3.0075,17.6164,263.0,0.1482,4.70,4.7876,343.544,
     1         1.9383,4.8536,5.9988,343.5416/
      DATA VT4/7.9925,3.0,279.7793,1.9383,4.8536,5.9988,343.5416,
     1         2.9169,7.1621,8.8500,343.8961/
      DATA xv03,yv03,zv03,xv04,yv04,zv04/-0.6883,-1.10615,29.0075,
     1                                  -0.6485,-2.49404,40.0075/
*
      DATA VT1/3.0075,18.0,269.9588,0.1482,4.70,4.7876,16.456,
     1         1.9383,4.8536,5.9988,16.4584/
      DATA VT2/7.9925,3.197,260.2207,1.9383,4.8536,5.9988,16.4584,
     1         2.9169,7.1621,8.8500,16.1039/
      DATA xv01,yv01,zv01,xv02,yv02,zv02/0.6883,-1.10615,29.0075,
     1                                  0.6485,-2.49404,40.0075/
*
**********************Define materials************************
*
      CALL GSMATE( 1,'AIR$',14.610,7.300,1.205e-03,3.04e+04,1.e+16,0,0)
      CALL GSMIXT( 2,'PLAST SC$',A1,Z1,DENS1,NLMAT1,WMAT1)
      CALL GSMIXT( 3,'CsI     $',A2,Z2,DENS2,NLMAT2,WMAT2)
      CALL GSMIXT( 4,'ACRYLIC$',A3,Z3,DENS3,NLMAT3,WMAT3)
      CALL GSMATE( 5,'CARBON$',12.010,6.0,2.265,18.8,0,0,0)
      CALL GSMIXT( 6,'MYLAR$',AMYL,ZMYL,DMYL,NMYL,WMYL)
      CALL GSMIXT( 7,'MAGIC GAS$',AGAS,ZGAS,DGAS,NGAS,WGAS)
      CALL GSMATE( 8,'TUNGSTEN$',183.850,74.0,19.3,0.35,0,0,0)
      CALL GSMATE( 9,'LEAD$',207.190,82.0,11.35,0.59,0,0,0)
      CALL GSMIXT(10,'CH2$',A1,Z1,0.935,-2,WCH2)
*
************************Define media***************************
*
        EPSIL=1.0e-04
        ISVOL=0
*
        CALL GSTMED( 1,'AIR$'    ,  1 ,ISVOL,0,
     1            0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      DO I=2,10
        CALL GSTMED( I,'CsI$'    ,  3 ,ISVOL,0,
     1            0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
      ENDDO
*
      CALL GSTMED( 11,'PLATIC SCINT$'    ,  2 ,ISVOL,0,
```

```
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 12,'PLATIC SCINT$'    ,  2 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      DO 50 I=13,81
        CALL GSTMED( I,'PLATIC SCINT$'    ,  2 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
50    CONTINUE
*
      CALL GSTMED( 82,'ACRYLIC$'    ,  4 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 83,'CARBON FIBER$'    ,  5 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 84,'MYLAR$'    ,  6 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 85,'MYLAR$'    ,  6 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 86,'MAGIC GAS$'    ,  7 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 87,'MAGIC GAS$'    ,  7 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 88,'TUNGSTEN$'    ,  8 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 89,'TUNGSTEN$'    ,  8 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      DO I=1,4
         K=I+89
        CALL GSTMED( K,'PLATIC SCINT$'    ,  2 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
      ENDDO
*
      CALL GSTMED( 94,'CH2$'    , 10 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 95,'CH2$'    ,  10 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      CALL GSTMED( 96,'LEAD$'    ,  9 ,ISVOL,0,
     1           0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
```

```
      CALL GSTMED( 97,'WAVE L. SHI$'    ,  2 ,ISVOL,0,
     1            0.0,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,0,0 )
*
      DO  k=1,69
        ACR(k) = 'C'//TAG(k)
      enddo
*
******************Defines geometry of the set-up******************
*
      CALL GSVOLU ('MAMA','BOX ',1,PAR,3,IVOL)     ! Experimental Area.
      CALL GSVOLU ('CALO','SPHE',1,CAL,6,ICVOL)    ! The Calorimeter.
      CALL GSVOLU ('TSTB','TUBE',12,TUB,3,ITVOL)   ! The Target.
      CALL GSVOLU ('PVET','TRAP',11,PLAS,11,IPV)   ! The Plastic Veto.
      CALL GSVOLU ('CFCY','TUBE',83,CFC,3,ICF)     ! The Carbon Fiber Cylinder.
      CALL GSVOLU ('ICWA','TUBE',84,WCI,3,IWI1)    ! MWPC1  Wall.
      CALL GSVOLU ('OCWA','TUBE',85,WCO,3,IWO1)     ! MWPC2  Wall.
      CALL GSVOLU ('GAS1','TUBE',86,WC1GAS,3,IGA1) ! MWPC1 GAS.
      CALL GSVOLU ('GAS2','TUBE',87,WC2GAS,3,IGA2) ! MWPC2 GAS.
      CALL GSVOLU ('ENCI','TUBE',94,ENCI,3,IENI)   ! End Cap Inner (read out).
      CALL GSVOLU ('ENCO','TUBE',95,ENCO,3,IENO)   ! End Cap Outer (read out).
      CALL GSVOLU ('WIR1','TUBE',88,AWIRE1,3,IWR1) ! Anode wire MWPC1.
      CALL GSVOLU ('WIR2','TUBE',89,AWIRE2,3,IWR2) ! Anode wire MWPC2.
      CALL GSVOLU ('CVFR','BOX ',90,CVF,3,ICVF)    ! Front Cosmic Veto.
      CALL GSVOLU ('CVBA','BOX ',91,CVB,3,ICVB)    ! Back Cosmic Veto.
      CALL GSVOLU ('CVTO','BOX ',92,CVT,3,ICVT)    ! Top Cosmic Veto.
      CALL GSVOLU ('CVSI','BOX ',93,CVS,3,ICVS)   ! Side Cosmic Veto.
      CALL GSVOLU ('LEAT','BOX ',96,TLEA,3,ILET)   ! Lead Top.
      CALL GSVOLU ('LEAF','BOX ',96,FLEA,3,ILEF)   ! Lead Front.
      CALL GSVOLU ('LEAB','BOX ',96,BLEA,3,ILEB)   ! Lead Back.
      CALL GSVOLU ('LEAS','BOX ',96,SLEA,3,ILS1)   ! Lead Sides.
      CALL GSVOLU ('WLSF','BOX ',97,WLSF,3,IWLF)   ! Wave length Shifter Front.
      CALL GSVOLU ('WLST','BOX ',97,WLST,3,IWLT)   ! Wave Length Shifter Top.
      CALL GSVOLU ('WLSB','BOX ',97,WLSB,3,IWLB)   ! Wave Length Shifter Back.
      CALL GSVOLU ('WLSS','BOX ',97,WLSS,3,IWLS)   ! Wave Length Shifter Sides.
*
************Define the 69 rectangular plastic fibers***********
*
      do  i=1,69
         CALL GSVOLU (TAG(i),'BOX ',(i+12),fiber,3,IVOLU)
      enddo
*
************Define the cladding around each fiber*****************
*
      DO j=1,69
         CALL GSVOLU (ACR(j),'BOX ',82,clad,3,IVOLU)
      enddo
*
*************Define the calorimeter modules**********************
*
```

```
      CALL GSVOLU ('PEN1','TRAP',2,PE1,11,IPEN1)
      CALL GSVOLU ('PEN2','TRAP',2,PE2,11,IPEN2)
*
      CALL GSVOLU ('HEA1','TRAP',3,HA1,11,IVOA1)
      CALL GSVOLU ('HEA2','TRAP',3,HA2,11,IVOA2)
*
      CALL GSVOLU ('HEB1','TRAP',4,HB1,11,IVOB1)
      CALL GSVOLU ('HEB2','TRAP',4,HB2,11,IVOB2)
*
      CALL GSVOLU ('HEC1','TRAP',5,HC1,11,IVOC1)
      CALL GSVOLU ('HEC2','TRAP',5,HC2,11,IVOC2)
*
      CALL GSVOLU ('HED1','TRAP',6,HD1,11,IVOD1)
      CALL GSVOLU ('HED2','TRAP',6,HD2,11,IVOD2)
*
      CALL GSVOLU ('HHD1','TRAP',7,HH1,11,IVOH1)
      CALL GSVOLU ('HHD2','TRAP',8,HH2,11,IVOH2)
*
      CALL GSVOLU ('VT11','TRAP',9,VT1,11,IVOV1)
      CALL GSVOLU ('VT12','TRAP',9,VT2,11,IVOV2)
      CALL GSVOLU ('VT21','TRAP',10,VT3,11,IVOV3)
      CALL GSVOLU ('VT22','TRAP',10,VT4,11,IVOV4)
*
***************Position the fibers in the target****************************
*
      k=1
      i=1
      do while(i.le.9)
         X=X0(i)
         do j=1,num(i)
           CALL GSPOS (TAG(k),1,ACR(k),0.0,0.0,0.0,0,'ONLY')
           CALL GSPOS (ACR(k),1,'TSTB',x,Y0(i),0.0,0,'ONLY')
           X=X+0.316
           k=k+1
          enddo
         i=i+1
      enddo
*
************************Position modules in the world*******************
*
      OPEN(UNIT=21,FILE='pent.dat',STATUS='OLD')
      OPEN(UNIT=22,FILE='hea.dat',STATUS='OLD')
      OPEN(UNIT=23,FILE='heb.dat',STATUS='OLD')
      OPEN(UNIT=24,FILE='hec.dat',STATUS='OLD')
      OPEN(UNIT=25,FILE='hed.dat',STATUS='OLD')
      OPEN(UNIT=26,FILE='hdh.dat',STATUS='OLD')
      OPEN(UNIT=27,FILE='vet.dat',STATUS='OLD')
*
      do i=1,10
```

```
         READ(21,80) TP11,PP11,TP12,PP12,TP13,PP13
80       FORMAT(2x,6f15.6)
*
         TP21=180.0-TP11
         PP21=PP11+180.0
         TP22=180-TP12
         PP22=PP12+180.0
         TP23=TP13
         PP23=PP13
         if (pp21.gt.360.0) pp21=pp21-360.0
         if (pp22.gt.360.0) pp22=pp22-360.0
*
         CALL GSROTM(i,TP11,PP11,TP12,PP12,TP13,PP13)
         CALL GSROTM(i+10,TP21,PP21,TP22,PP22,TP23,PP23)
*
         xp1=xp01*sind(tp11)*cosd(pp11)+yp01*sind(tp12)*
     1    cosd(pp12)+zp01*sind(tp13)*cosd(pp13)
         yp1=xp01*sind(tp11)*sind(pp11)+yp01*sind(tp12)*
     1    sind(pp12)+zp01*sind(tp13)*sind(pp13)
         zp1=xp01*cosd(tp11)+yp01*cosd(tp12)+zp01*cosd(tp13)
*
         xp2=xp02*sind(tp21)*cosd(pp21)+yp02*sind(tp22)*
     1    cosd(pp22)+zp02*sind(tp23)*cosd(pp23)
         yp2=xp02*sind(tp21)*sind(pp21)+yp02*sind(tp22)*
     1    sind(pp22)+zp02*sind(tp23)*sind(pp23)
         zp2=xp02*cosd(tp21)+yp02*cosd(tp22)+zp02*cosd(tp23)
*
         CALL GSPOS ('PEN1',i,'CALO',xp1,yp1,zp1,i,'ONLY')
         CALL GSPOS ('PEN2',i,'CALO',xp2,yp2,zp2,i+10,'ONLY')
      enddo
*
       do i=21,70
         READ(22,100) TA11,PA11,TA21,PA21,TA31,PA31
         READ(22,100) TA12,PA12,TA22,PA22,TA32,PA32
*
         READ(23,100) TB11,PB11,TB21,PB21,TB31,PB31
         READ(23,100) TB12,PB12,TB22,PB22,TB32,PB32
*
         READ(24,100) TC11,PC11,TC21,PC21,TC31,PC31
         READ(24,100) TC12,PC12,TC22,PC22,TC32,PC32
*
      if (i.le.60) then
         READ(25,100) TD11,PD11,TD21,PD21,TD31,PD31
         READ(25,100) TD12,PD12,TD22,PD22,TD32,PD32
      else
         READ(26,100) TD11,PD11,TD21,PD21,TD31,PD31
         READ(26,100) TD12,PD12,TD22,PD22,TD32,PD32
         READ(27,100) TV11,PV11,TV21,PV21,TV31,PV31
         READ(27,100) TV12,PV12,TV22,PV22,TV32,PV32
```

```
          tv11=180.0-tv11
          pv11=180.0+pv11
          tv21=180.0-tv21
          pv21=180.0+pv21
*
          tv12=180.0-tv12
          pv12=180.0+pv12
          tv22=180.0-tv22
          pv22=180.0+pv22
        endif
*
100     format(2x,6f15.6)
*
        xa1=xa01*sind(ta11)*cosd(pa11)+ya01*sind(ta21)*
     1    cosd(pa21)+za01*sind(ta31)*cosd(pa31)
        ya1=xa01*sind(ta11)*sind(pa11)+ya01*sind(ta21)*
     1    sind(pa21)+za01*sind(ta31)*sind(pa31)
        za1=xa01*cosd(ta11)+ya01*cosd(ta21)+za01*cosd(ta31)
*
        xa2=xa02*sind(ta12)*cosd(pa12)+ya02*sind(ta22)*
     1    cosd(pa22)+za02*sind(ta32)*cosd(pa32)
        ya2=xa02*sind(ta12)*sind(pa12)+ya02*sind(ta22)*
     1    sind(pa22)+za02*sind(ta32)*sind(pa32)
        za2=xa02*cosd(ta12)+ya02*cosd(ta22)+za02*cosd(ta32)
*
        xb1=xb01*sind(tb11)*cosd(pb11)+yb01*sind(tb21)*
     1    cosd(pb21)+zb01*sind(tb31)*cosd(pb31)
        yb1=xb01*sind(tb11)*sind(pb11)+yb01*sind(tb21)*
     1    sind(pb21)+zb01*sind(tb31)*sind(pb31)
        zb1=xb01*cosd(tb11)+yb01*cosd(tb21)+zb01*cosd(tb31)
*
        xb2=xb02*sind(tb12)*cosd(pb12)+yb02*sind(tb22)*
     1    cosd(pb22)+zb02*sind(tb32)*cosd(pb32)
        yb2=xb02*sind(tb12)*sind(pb12)+yb02*sind(tb22)*
     1    sind(pb22)+zb02*sind(tb32)*sind(pb32)
        zb2=xb02*cosd(tb12)+yb02*cosd(tb22)+zb02*cosd(tb32)
*
        xc1=xc01*sind(tc11)*cosd(pc11)+yc01*sind(tc21)*
     1    cosd(pc21)+zc01*sind(tc31)*cosd(pc31)
        yc1=xc01*sind(tc11)*sind(pc11)+yc01*sind(tc21)*
     1    sind(pc21)+zc01*sind(tc31)*sind(pc31)
        zc1=xc01*cosd(tc11)+yc01*cosd(tc21)+zc01*cosd(tc31)
*
        xc2=xc02*sind(tc12)*cosd(pc12)+yc02*sind(tc22)*
     1    cosd(pc22)+zc02*sind(tc32)*cosd(pc32)
        yc2=xc02*sind(tc12)*sind(pc12)+yc02*sind(tc22)*
     1    sind(pc22)+zc02*sind(tc32)*sind(pc32)
        zc2=xc02*cosd(tc12)+yc02*cosd(tc22)+zc02*cosd(tc32)
*
```

```
      if (i.gt.60) then
        yd01=-2.2295
        yd02=-2.2295
        xd01=-0.1117
        xd02=0.1117
        endif
      xd1=xd01*sind(td11)*cosd(pd11)+yd01*sind(td21)*
     1    cosd(pd21)+zd01*sind(td31)*cosd(pd31)
      yd1=xd01*sind(td11)*sind(pd11)+yd01*sind(td21)*
     1    sind(pd21)+zd01*sind(td31)*sind(pd31)
      zd1=xd01*cosd(td11)+yd01*cosd(td21)+zd01*cosd(td31)
*
      xd2=xd02*sind(td12)*cosd(pd12)+yd02*sind(td22)*
     1    cosd(pd22)+zd02*sind(td32)*cosd(pd32)
      yd2=xd02*sind(td12)*sind(pd12)+yd02*sind(td22)*
     1    sind(pd22)+zd02*sind(td32)*sind(pd32)
      zd2=xd02*cosd(td12)+yd02*cosd(td22)+zd02*cosd(td32)
*
      j=i+50
      if (i.gt.60) then
          k=i-60
          CALL GSROTM(k+420,TV11,PV11,TV21,PV21,TV31,PV31)
          CALL GSROTM(k+430,TV12,PV12,TV22,PV22,TV32,PV32)
*
          xv1=xv01*sind(tv11)*cosd(pv11)+yv01*sind(tv21)*
     1         cosd(pv21)+zv01*sind(tv31)*cosd(pv31)
          yv1=xv01*sind(tv11)*sind(pv11)+yv01*sind(tv21)*
     1         sind(pv21)+zv01*sind(tv31)*sind(pv31)
          zv1=xv01*cosd(tv11)+yv01*cosd(tv21)+zv01*cosd(tv31)
*
          xv2=xv02*sind(tv11)*cosd(pv11)+yv02*sind(tv21)*
     1         cosd(pv21)+zv02*sind(tv31)*cosd(pv31)
          yv2=xv02*sind(tv11)*sind(pv11)+yv02*sind(tv21)*
     1         sind(pv21)+zv02*sind(tv31)*sind(pv31)
          zv2=xv02*cosd(tv11)+yv02*cosd(tv21)+zv02*cosd(tv31)
*
          xv3=xv03*sind(tv12)*cosd(pv12)+yv03*sind(tv22)*
     1         cosd(pv22)+zv03*sind(tv32)*cosd(pv32)
          yv3=xv03*sind(tv12)*sind(pv12)+yv03*sind(tv22)*
     1         sind(pv22)+zv03*sind(tv32)*sind(pv32)
          zv3=xv03*cosd(tv12)+yv03*cosd(tv22)+zv03*cosd(tv32)
*
          xv4=xv04*sind(tv12)*cosd(pv12)+yv04*sind(tv22)*
     1         cosd(pv22)+zv04*sind(tv32)*cosd(pv32)
          yv4=xv04*sind(tv12)*sind(pv12)+yv04*sind(tv22)*
     1         sind(pv22)+zv04*sind(tv32)*sind(pv32)
          zv4=xv04*cosd(tv12)+yv04*cosd(tv22)+zv04*cosd(tv32)
*
      endif
```

```
*
      CALL GSROTM(i,TA11,PA11,TA21,PA21,TA31,PA31)
      CALL GSROTM(j,TA12,PA12,TA22,PA22,TA32,PA32)
*
      CALL GSROTM(i+100,TB11,PB11,TB21,PB21,TB31,PB31)
      CALL GSROTM(j+100,TB12,PB12,TB22,PB22,TB32,PB32)
*
      CALL GSROTM(i+200,TC11,PC11,TC21,PC21,TC31,PC31)
      CALL GSROTM(j+200,TC12,PC12,TC22,PC22,TC32,PC32)
*
      CALL GSROTM(i+300,TD11,PD11,TD21,PD21,TD31,PD31)
      CALL GSROTM(j+300,TD12,PD12,TD22,PD22,TD32,PD32)
*
      ia=i-20
      CALL GSPOS ('HEA1',ia,'CALO',xa1,ya1,za1,i,'ONLY')
      CALL GSPOS ('HEA2',ia,'CALO',xa2,ya2,za2,j,'ONLY')
*
      ib=i-20
      CALL GSPOS ('HEB1',ib,'CALO',xb1,yb1,zb1,i+100,'ONLY')
      CALL GSPOS ('HEB2',ib,'CALO',xb2,yb2,zb2,j+100,'ONLY')
*
      ic=i-20
      CALL GSPOS ('HEC1',ic,'CALO',xc1,yc1,zc1,i+200,'ONLY')
      CALL GSPOS ('HEC2',ic,'CALO',xc2,yc2,zc2,j+200,'ONLY')
*
      if (i.le.60) then
        id=i-20
        CALL GSPOS ('HED1',id,'CALO',xd1,yd1,zd1,i+300,'ONLY')
        CALL GSPOS ('HED2',id,'CALO',xd2,yd2,zd2,j+300,'ONLY')
      else
        ih=i-60
        iv=i-60
        CALL GSPOS ('HHD1',ih,'CALO',xd1,yd1,zd1,i+300,'ONLY')
        CALL GSPOS ('HHD2',ih,'CALO',xd2,yd2,zd2,j+300,'ONLY')
        CALL GSPOS ('VT11',iv,'CALO',xv1,yv1,zv1,iv+420,'ONLY')
        CALL GSPOS ('VT12',iv,'CALO',xv2,yv2,zv2,iv+420,'ONLY')
        CALL GSPOS ('VT21',iv,'CALO',xv3,yv3,zv3,iv+430,'ONLY')
        CALL GSPOS ('VT22',iv,'CALO',xv4,yv4,zv4,iv+430,'ONLY')
      endif
*
      enddo
*
****************Position the 20 plastic veto staves ****************
*
      R0 = 13.15875
      Z0 = 0.0
      ID=500
      DO I =1,20
         PSI = 18.0*(I-1)
```

```
         THY = 90.0
         PHY = PSI
         THX = 90.0
         PHX = 90 + PSI
         IF (PHX.GT.360) PHX = PHX-360
         THZ = 0.0
         PHZ = 0.0
         CALL GSROTM(ID,THX,PHX,THY,PHY,THZ,PHZ)
         XPV = R0*COSD(PSI)
         YPV = R0*SIND(PSI)
         ZPV = Z0
         CALL GSPOS('PVET',I,'MAMA',XPV,YPV,ZPV,ID,'ONLY')
         ID = ID + 1
      ENDDO
*
***************Position the carbon fiber cylinder*****************
*
      CALL GSPOS('CFCY',1,'MAMA',0,0,0,0,'ONLY')
*
***************Position the anode wires of MWPC 1*****************
*
      R0 = 6.015
      Z0 = 0.0
      DO K=1,192
         PSI = 360.0*(K-1)/192
         XW1 = R0*COSD(PSI)
         YW1 = R0*SIND(PSI)
         ZW1 = Z0
         CALL GSPOS('WIR1',K,'GAS1',XW1,YW1,ZW1,0,'ONLY')
      ENDDO
*
***************Position the anode wires of MWPC 2******************
*
      R0 = 12.01
      Z0 = 0.0
      DO K=1,385
         PSI = 360.0*(K-1)/385
         XW2 = R0*COSD(PSI)
         YW2 = R0*SIND(PSI)
         ZW2 = Z0
         CALL GSPOS('WIR2',K,'GAS2',XW2,YW2,ZW2,0,'ONLY')
      ENDDO
*
***************Position MWPC1 gas and MWPC2 gas*********************
*
      CALL GSPOS('GAS1',1,'ICWA',0,0,0,0,'ONLY')
      CALL GSPOS('GAS2',1,'OCWA',0,0,0,0,'ONLY')
*
**************Position the MWPC walls******************************
```

```
*
      CALL GSPOS('ICWA',1,'MAMA',0,0,0,0,'ONLY')
      CALL GSPOS('OCWA',1,'MAMA',0,0,0,0,'ONLY')
*
**************Position the end caps for MWPC read out***************
*
      CALL GSPOS ('ENCI',1,'MAMA',0,0,22.5,0,'ONLY')
      CALL GSPOS ('ENCI',2,'MAMA',0,0,-22.5,0,'ONLY')
      CALL GSPOS ('ENCO',1,'MAMA',0,0,32.0,0,'ONLY')
      CALL GSPOS ('ENCO',2,'MAMA',0,0,-32.0,0,'ONLY')
*
***************Position the target and the calorimeter***************
*
      CALL GSPOS ('TSTB',1,'MAMA',0.0,0.0,0.0,0,'ONLY')
      CALL GSPOS ('CALO',1,'MAMA',0.0,0.0,0.0,0,'ONLY')
*
***************Position the lead walls*******************************
*
      CALL GSPOS ('LEAT',1,'MAMA',0,121.2,25.48,0,'ONLY')
      CALL GSPOS ('LEAF',1,'MAMA',0,-2.5,-110.46,0,'ONLY')
      CALL GSPOS ('LEAB',1,'MAMA',0,15.05,161.42,0,'ONLY')
      CALL GSPOS ('LEAS',1,'MAMA',129.54,15.05,25.48,0,'ONLY')
      CALL GSPOS ('LEAS',2,'MAMA',-129.54,15.05,25.48,0,'ONLY')
*
****************Position the cosmic vetoes***************************
*
      CALL GSPOS ('CVTO',1,'MAMA',0,127.51,25.48,0,'ONLY')
      CALL GSPOS ('CVFR',1,'MAMA',0,0,-114.23,0,'ONLY')
      CALL GSPOS ('CVBA',1,'MAMA',0,15.05,165.19,0,'ONLY')
      CALL GSPOS ('CVSI',1,'MAMA',133.31,18.8,25.48,0,'ONLY')
      CALL GSPOS ('CVSI',2,'MAMA',-133.31,18.8,25.48,0,'ONLY')
*
***************Position the Wave Length Shifters************************
*
      CALL GSPOS ('WLSF',1,'MAMA',0,124.97,-114.23,0,'ONLY')
      CALL GSPOS ('WLSF',2,'MAMA',0,-124.97,-114.23,0,'ONLY')
      CALL GSPOS ('WLSB',1,'MAMA',0,124.97,165.19,0,'ONLY')
      CALL GSPOS ('WLSB',2,'MAMA',0,-94.87,165.19,0,'ONLY')
      CALL GSPOS ('WLST',1,'MAMA',130.77,127.51,25.48,0,'ONLY')
      CALL GSPOS ('WLST',2,'MAMA',-130.77,127.51,25.48,0,'ONLY')
      CALL GSPOS ('WLSS',1,'MAMA',133.31,132.47,25.48,0,'ONLY')
      CALL GSPOS ('WLSS',2,'MAMA',133.31,-94.87,25.48,0,'ONLY')
      CALL GSPOS ('WLSS',3,'MAMA',-133.31,132.47,25.48,0,'ONLY')
      CALL GSPOS ('WLSS',4,'MAMA',-133.31,-94.87,25.48,0,'ONLY')
*
***************Close data files*************************************
*
      close(21)
      close(22)
```

```
      close(23)
      close(24)
      close(25)
      close(26)
      close(27)
*
******************Close the geometry package************************
*
      CALL GGCLOS
*
*     PRINT INFORMATION OF GEOMETRY OF THE EXPERIMENTAL SETUP
*
*      CALL GPVOLU(0)
*      CALL GPMATE(0)
*      CALL GPTMED(0)
*
      END
*
*CMZ :          30/05/89  09.13.13  by  Rene Brun
*
*-- Author :    DEC-16-91 KETEVI A. ASSAMAGAN
*
      SUBROUTINE GUKINE
*
**********************************************************************
*                                                                  *
*             Read or Generates Kinematics for primary tracks        *
*                                                                  *
**********************************************************************
*
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     1        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
C
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     1      ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
*
      COMMON/energies/calo1(240),calo2(240),calo3(240),tgt1(4),
     1               tgt2(4),tgt3(4),stf(3),tpv1(20),tpv2(20),
     1               tpv3(20),tcv1(5),tcv2(5),tcv3(5)
*
      COMMON/delali/pent(10),hexa(50),hexb(50),hexc(50),
     1             hexd(40),hdh(20),veto(20),etarg,ewhole,
     1             epveto,etgt,eclad,SQMA,W,etot,pwhole,
     1             pes(240),pnu(240),pv(20),cv(5),wcx(2),
     1             wcy(2),wcz(2)
*
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,100),TOFD(100),IFLGK(100)
      COMMON/GCNUM/NMATE,NVOLUM,NROTM,NTMED,NTMULT,NTRACK,NPART,
     1            NSTMAX,NVERTX,NHEAD,NBIT
```

```
*
      COMMON/FLAG/iflag,iad,jcount,vertex(3)
      VOLATILE/flag/
      COMMON/MAWUSE/E1,X1,Y1,Z1,E2,X2,Y2,Z2,E3,X3,Y3,Z3
*
      COMMON/ketevi/thetal,phil,pgam(8),thetag(2),phig(2)
     1              ,plpi0,elpi0,relang,beta,gamma
*
      VOLATILE/ketevi/
      VOLATILE/MAWUSE/
      REAL*4 PLAB(3)
      logical check1,check2
      INTEGER ik
      DATA ISEED/98793/
      DATA iflag,jcount/0,0/
*
*     -----------------------------------------------------------------
*
      IFLAG = IFLAG + 1
      IAD = IFLAG - 3*jcount
*
      IF (IAD.eq.1) THEN
*
 2      continue
        CALL DECAY3B(W,SQMA)
*         if (W.lt.RAN(iseed)) goto 2
*
        plpi0  = sqrt(x1*x1 + y1*y1 + z1*z1)
        elpi0  = E1
        thetal = acosd(z1/plpi0)
        phil   = atan2d(y1,x1)
        CALL DECAY_PI0
*
        p1=pgam(4)
        p2=pgam(4)
 10     continue
        th1=180.0*ran(iseed)
        th2=relang-th1
        if (th2.lt.0) goto 10
        ph1=360*ran(iseed)
        ph2=180.0+ph1
        if (ph2.gt.360.0) ph2=ph2-360.0
*
        pgam(1)=p1*sind(th1)*cosd(ph1)
        pgam(2)=p1*sind(th1)*sind(ph1)
        pgam(3)=p1*cosd(th1)
        pgam(5)=p2*sind(th2)*cosd(ph2)
        pgam(6)=p2*sind(th2)*sind(ph2)
        pgam(7)=p2*cosd(th2)
```

```
*
        IKINE = 1
        PLAB(1) = pgam(1)
        PLAB(2) = pgam(2)
        PLAB(3) = pgam(3)
*
        read (20) vertex(1),vertex(2),vertex(3)
*
      ELSEIF (IAD.EQ.2) THEN
        PLAB(1) = pgam(5)
        PLAB(2) = pgam(6)
        PLAB(3) = pgam(7)
        IKINE = 1
      ELSE
        PLAB(1) = X2
        PLAB(2) = Y2
        PLAB(3) = Z2
        IKINE = 2
      ENDIF
      ik = IFLAG/3
      IF ((IFLAG-3*ik).EQ.0) THEN
          jcount = jcount + 1
      ENDIF
*
      CALL GSVERT(VERTEX,0,0,0,0,NVERT)
      CALL GSKINE(PLAB,IKINE,NVERT,0,0,NT)
      END
*
*CMZ :          13/05/89  19.04.22  by  Rene Brun
*-- Author :
      SUBROUTINE GUTREV
*
************************************************************************
*                                                                    *
*             User routine to control tracking of one event            *
*                                                                    *
*             Called by GRUN                                          *
*                                                                    *
************************************************************************
*
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     1      ,DEEMAX,EPSIL,STMIN,CFIELD,PREC,IUPD,ISTPAR,NUMOLD
*
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(30),NAMEC(30)
     1 ,NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG ,STEP  ,SNEXT ,SFIELD
     1 ,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL,ISTOP ,IGAUTO,IEKBIN
     1 , ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN,NLVSAV,ISTORY
*
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,100),TOFD(100),IFLGK(100)
```

```
*
      COMMON/GCNUM/NMATE,NVOLUM,NROTM,NTMED,NTMULT,NTRACK,NPART,
     1            NSTMAX,NVERTX,NHEAD,NBIT
*
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     1             LVOLUM(15),LINDEX(15),INFROM,NLEVMX,
     1             NLDEV(15),LINMX(15),GTRAN(3,15),GRMAT(10,15),
     1             GONLY(15),GLX(3)
*
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
*
*
*     ------------------------------------------------------------------
*
      MAXNST = 100000
*
      CALL GTREVE
*
      END
*CMZ :          13/05/89  19.04.22  by  Rene Brun
*-- Author :  KETEVI A. ASSAMAGAN
*
      SUBROUTINE GUSTEP
*
************************************************************************
*                                                                    *
*             User routine called at the end of each tracking step     *
*             MEC   is the mechanism origin of the step               *
*             INWVOL is different from 0 when the track has reached    *
*                    a volume boundary                               *
*             ISTOP is different from 0 if the track has stopped      *
*                                                                    *
************************************************************************
*
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     1      ,DEEMAX,EPSIL,STMIN,CFIELD,PREC,IUPD,ISTPAR,NUMOLD

*
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     1      ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
*
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,100),TOFD(100),IFLGK(100)
*
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     1        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
*
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(30),NAMEC(30)
     1 ,NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG ,STEP  ,SNEXT ,SFIELD
```

```
     1 ,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL,ISTOP ,IGAUTO,IEKBIN
     1 , ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN,NLVSAV,ISTORY
*
      COMMON/GCNUM/NMATE,NVOLUM,NROTM,NTMED,NTMULT,NTRACK,NPART,
     1             NSTMAX,NVERTX,NHEAD,NBIT
*
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     1              LVOLUM(15),LINDEX(15),INFROM,NLEVMX,
     1              NLDEV(15),LINMX(15),GTRAN(3,15),GRMAT(10,15),
     1              GONLY(15),GLX(3)
*
      COMMON/energies/calo1(240),calo2(240),calo3(240),tgt1(4),
     1                tgt2(4),tgt3(4),stf(3),tpv1(20),tpv2(20),
     1                tpv3(20),tcv1(5),tcv2(5),tcv3(5)
*
      COMMON/delali/pent(10),hexa(50),hexb(50),hexc(50),
     1              hexd(40),hdh(20),veto(20),etarg,ewhole,
     1              epveto,etgt,eclad,SQMA,W,etot,pwhole,
     1              pes(240),pnu(240),pv(20),cv(5),wcx(2),
     1              wcy(2),wcz(2)
*
      COMMON/FLAG/IFLAG,IAD,jcount,vertex(3)
      VOLATILE/FLAG/
      LOGICAL CHECK
*
      REAL MW11(6),MW12(6),XINT(6),PZINT(4)
      EQUIVALENCE (MW11(1),VECT(1))
      EQUIVALENCE (MW12(1),VOUT(1))
*
*
*     -----------------------------------------------------------------
*
*****************MWPC hits by Emil Frlez 06-13-95**********************
*
      RHOMW1=SQRT(VECT(1)**2+VECT(2)**2)
*
C hits in first MWPC, efficiency 0.97
*
      IF(ABS(VECT(3)).LE.17.5.AND.ABS(RHOMW1-6.015).LE.0.275
     +   .AND.CHARGE.NE.0.) THEN ! track-cylinder intersection MWFLAG=1
         CALL GICYL(6.015,MW11,MW12,0.1,0.1,2,XINT,SINT,
     +              PZINT,MWFLAG)
         IF(MWFLAG.EQ.1.AND.RNDM(ISEED).LT.0.97) THEN
            CALL GRANOR(RAN1,RAN2)
            CALL GRANOR(RAN3,RAN4)
            WCX(1)=XINT(1)*(1.+0.085*RAN1)
            WCY(1)=XINT(2)*(1.+0.085*RAN2)
            WCZ(1)=XINT(3)*(1.+0.085*RAN3)
         ENDIF
```

```
      ENDIF
*
C hits in second MWPC, efficiency 0.97
*
      IF(ABS(VECT(3)).LE.27..AND.ABS(RHOMW1-12.0).LE.0.275
     +   .AND.CHARGE.NE.0.) THEN
         CALL GICYL(12.01,MW11,MW12,0.1,0.1,2,XINT,SINT,
     +              PZINT,MWFLAG)
         IF(MWFLAG.EQ.1.AND.RNDM(ISEED).LT.0.97) THEN
            CALL GRANOR(RAN1,RAN2)
            CALL GRANOR(RAN3,RAN4)
            WCX(2)=XINT(1)*(1.+0.085*RAN1)
            WCY(2)=XINT(2)*(1.+0.085*RAN2)
            WCZ(2)=XINT(3)*(1.+0.085*RAN3)
         ENDIF
      ENDIF
*
******Collect energies dumped in various detectors KAA 06-13-95*********
*
      CHECK = NUMED.EQ.13
      DO i=14,81
       CHECK = CHECK.OR.(NUMED.EQ.i)
      ENDDO
*
      if (destep.ne.0.0) then
*
         if (iad.le.2) then
*
            ewhole=ewhole+1000.0*destep ! 2 gamma energies
*
            if ((numed.ge.2).and.(numed.le.8)) then
                etot=etot+1000.0*destep ! 220 CsI active elements
            endif
*
         else
*
             pwhole=pwhole+1000.0*destep  ! Total positron energy
*
         endif
*
         if (numed.eq.12) etarg=etarg+1000.0*destep ! Guard ring elements!
*
         if (numed.eq.11) then
            epveto=epveto+1000.0*destep ! Total plastic veto energy.
            do j=1,20
               if (number(2).eq.j) then
                  pv(j) = pv(j)+1000*destep ! Plastic veto staves
               endif
            enddo
```

```
         endif
*
         if (check) etgt=etgt+1000.0*destep ! 69 fiber energies
*
         if (numed.eq.82) eclad=eclad+1000.0*destep ! Energy lost in cladding!
*
      endif
*
*         Collect energy deposited in each of 240 CsI elements:
*
      if (numed.eq.2) then
        do j=1,10
         if (number(3).eq.j) then
            pent(j)=pent(j)+1000.0*destep ! The 10 pentagons
         endif
        enddo
      endif
*
      if (numed.eq.3) then
         do j=1,50
          if (number(3).eq.j) then
             hexa(j)=hexa(j)+1000.0*destep ! 50 hexagon A's
          endif
         enddo
      endif
*
      if (numed.eq.4) then
         do j=1,50
           if (number(3).eq.j) then
              hexb(j)=hexb(j)+1000.0*destep ! 50 hexagon B's
           endif
          enddo
      endif
*
      if (numed.eq.5) then
        do j=1,50
          if (number(3).eq.j) then
             hexc(j)=hexc(j)+1000.0*destep ! 50 hexagon C's
          endif
         enddo
       endif
*
       if (numed.eq.6) then
         do i=1,40
           if (number(3).eq.i) then
             hexd(i)=hexd(i)+1000.0*destep ! 40 hexagon D's
           endif
          enddo
        endif
```

```
*
       if (numed.eq.7) then
          do j=1,10
            k=j-1
            h=2*k+1
            if (number(3).eq.j) then
               hdh(h)=hdh(h)+1000.0*destep ! 10 HD1
            endif
          enddo
       endif
*
       if (numed.eq.8) then
          do j=11,20
            k=j-10
            h=2*k
            if (number(3).eq.(j-10)) then
               hdh(h)=hdh(h)+1000.0*destep ! 10 HD2
            endif
           enddo
        endif
*
        if (numed.eq.9) then
           do j=1,10
             k=j-1
             h=2*k+1
             if (number(3).eq.j) then
               veto(h)=veto(h)+1000.0*destep ! 10 CsI veto 1
             endif
            enddo
         endif
*
        if (numed.eq.10) then
           do j=11,20
             k=j-10
             h=2*k
             if (number(3).eq.(j-10)) then
               veto(h)=veto(h)+1000.0*destep ! 10 CsI veto 2
             endif
            enddo
         endif
*
*        The cosmic veto detectors
*
         if (numed.eq.90) cv(1) = cv(1)+1000*destep ! Front cosmic veto
         if (numed.eq.91) cv(2) = cv(2)+1000*destep ! Back cosmic veto
         if (numed.eq.92) cv(3) = cv(3)+1000*destep ! Top cosmic veto
         if (numed.eq.93) then
            do j=1,2
               if (number(2).eq.j) cv(3+j) = cv(3+j)+1000*destep ! Side C. vetoes.
```

```
           enddo
         endif
*
      IF(NGKINE.GT.0) THEN
        DO 5 I=1,NGKINE
          ITYPA  = GKIN(5,I)
          IF(ITYPA.NE.4) CALL GSKING(I)
   5    CONTINUE
      ENDIF
*
*             Debug/plot event
*
*      CALL GDCXYZ
*
      END
*CMZ :          02/03/89  08.37.38  by  Rene Brun
*
*-- Author :    DEC-16-91 KETEVI A. ASSAMAGAN
*
      SUBROUTINE GUOUT
*
C    ****************************************************************
C    *                                                              *
C    *       User routine called at the end of each event.          *
C    *                                                              *
C    ****************************************************************
*
      COMMON/GCUNIT/LIN,LOUT,NUNITS,LUNITS(5)
      INTEGER LIN,LOUT,NUNITS,LUNITS
      COMMON/GCMAIL/CHMAIL
      CHARACTER*132 CHMAIL
*
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     1      ,DEEMAX,EPSIL,STMIN,CFIELD,PREC,IUPD,ISTPAR,NUMOLD
*
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,100),TOFD(100),IFLGK(100)
*
      COMMON/GCNUM/NMATE,NVOLUM,NROTM,NTMED,NTMULT,NTRACK,NPART,
     1             NSTMAX,NVERTX,NHEAD,NBIT
*
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     1              LVOLUM(15),LINDEX(15),INFROM,NLEVMX,
     1              NLDEV(15),LINMX(15),GTRAN(3,15),GRMAT(10,15),
     1              GONLY(15),GLX(3)
*
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(30),NAMEC(30)
     1 ,NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG ,STEP  ,SNEXT ,SFIELD
     1 ,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL,ISTOP ,IGAUTO,IEKBIN
     1 , ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN,NLVSAV,ISTORY
```

```
*
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     1        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
*
      COMMON/energies/calo1(240),calo2(240),calo3(240),tgt1(4),
     1               tgt2(4),tgt3(4),stf(3),tpv1(20),tpv2(20),
     1               tpv3(20),tcv1(5),tcv2(5),tcv3(5)
*
      COMMON/delali/pent(10),hexa(50),hexb(50),hexc(50),
     1             hexd(40),hdh(20),veto(20),etarg,ewhole,
     1             epveto,etgt,eclad,SQMA,W,etot,pwhole,
     1             pes(240),pnu(240),pv(20),cv(5),wcx(2),
     1             wcy(2),wcz(2)
*
      COMMON/ketevi/thetal,phil,pgam(8),thetag(2),phig(2)
     1             ,plpi0,elpi0,relang,beta,gamma
*
      COMMON/FLAG/IFLAG,IAD,jcount,vertex(3)
      VOLATILE/FLAG/
      VOLATILE/ketevi/
      REAL*4 POS(287),eint(240),sig(240)
*
      nc = 37895
*
      if (iad.eq.1) then
*
         do i=1,10
            calo1(i)=pent(i)
         enddo
         do i=11,60
            calo1(i)=hexa(i-10)
         enddo
         do i=61,110
            calo1(i)=hexb(i-60)
         enddo
         do i=111,160
            calo1(i)=hexc(i-110)
         enddo
         do i=161,200
            calo1(i)=hexd(i-160)
         enddo
         do i=201,220
            calo1(i)=hdh(i-200)
         enddo
         do i=221,240
            calo1(i)=veto(i-220)
         enddo
*
         tgt1(1)=etgt
```

```
        tgt1(2)=eclad
        tgt1(3)=etarg
        tgt1(4)=epveto
*
        do k=1,20
           tpv1(k) = pv(k)
        enddo
*
        do k=1,5
           tcv1(k) = cv(k)
        enddo
*
        stf(1)=relang
        stf(2)=w
        stf(3)=sqma
*
      else if (iad.eq.2) then
*
        do i=1,10
           calo2(i)=pent(i)
        enddo
        do i=11,60
           calo2(i)=hexa(i-10)
        enddo
        do i=61,110
           calo2(i)=hexb(i-60)
        enddo
        do i=111,160
           calo2(i)=hexc(i-110)
        enddo
        do i=161,200
           calo2(i)=hexd(i-160)
        enddo
        do i=201,220
           calo2(i)=hdh(i-200)
        enddo
        do i=221,240
           calo2(i)=veto(i-220)
        enddo
*
        tgt2(1)=etgt
        tgt2(2)=eclad
        tgt2(3)=etarg
        tgt2(4)=epveto
*
        do k=1,20
           tpv2(k) = pv(k)
        enddo
*
```

```
         do k=1,5
            tcv2(k) = cv(k)
         enddo
*
      else
         do i=1,10
            calo3(i)=pent(i)
         enddo
         do i=11,60
            calo3(i)=hexa(i-10)
         enddo
         do i=61,110
            calo3(i)=hexb(i-60)
         enddo
         do i=111,160
            calo3(i)=hexc(i-110)
         enddo
         do i=161,200
            calo3(i)=hexd(i-160)
         enddo
         do i=201,220
            calo3(i)=hdh(i-200)
         enddo
         do i=221,240
            calo3(i)=veto(i-220)
         enddo
*
         tgt3(1)=etgt
         tgt3(2)=eclad
         tgt3(3)=etarg
         tgt3(4)=epveto
*
         do k=1,20
            tpv3(k) = pv(k)
         enddo
*
         do k=1,5
            tcv3(k) = cv(k)
         enddo
*
         POS(1)=tgt1(1)+tgt2(1)
         POS(2)=tgt1(2)+tgt2(2)
         POS(3)=tgt1(4)+tgt2(4)
         POS(4)=tgt1(3)+tgt2(3)
*
         do i=1,240
            pos(i+4)=calo1(i)+calo2(i)+calo3(i)
         enddo
*
```

```
*         do k=1,240
*            if (eint(k).gt.0) then
*               sig(k)=(eint(k)/pes(k)+(eint(k)*pnu(k))**2)
*               sig(k) = sqrt(sig(k))
*               pos(k+4) = eint(k)+sig(k)*gasdev(nc)
*            endif
*         enddo
*
        POS(245)=etot
        POS(246)=ewhole
        POS(247)=stf(2)
        POS(248)=stf(1)
        POS(249)=tgt3(1)
        pos(250)=tgt3(2)
        pos(251)=tgt3(4)
        pos(252)=tgt3(3)
        pos(253)=pwhole
        pos(254)=stf(3)
        pos(255)=1000*pgam(4)
        pos(256)=1000*pgam(8)
*
        do k=1,20
           pos(256+k) = tpv1(k)+tpv2(k)+tpv3(k)
        enddo
        do k=1,5
           pos(276+k) = tcv1(k)+tcv2(k)+tcv3(k)
        enddo
*
        do k=1,2
           pos(281+k) = wcx(k)
           pos(283+k) = wcy(k)
           pos(285+k) = wcz(k)
        enddo
*
        call hfn(9,pos)
*
        do i=1,287
           pos(i)=0.0
        enddo
*
        do i=1,240
           calo1(i)=0.0
           calo2(i)=0.0
           calo3(i)=0.0
           eint(i)=0.0
           sig(i)=0.0
        enddo
*
        do i=1,4
```

```
            tgt1(i)=0.0
            tgt2(i)=0.0
            tgt3(i)=0.0
         enddo
         do i=1,3
            stf(i)=0.0
         enddo
*
         etot=0.0
         ewhole=0.0
         pwhole=0.0
*
         do k=1,2
            wcx(k) = 0.0
            wcy(k) = 0.0
            wcz(k) = 0.0
         enddo
*
         do k=1,20
            tpv1(k) = -100.0
            tpv2(k) = -100.0
            tpv3(k) = -100.0
         enddo
*
         do k=1,5
            tcv1(k) = 0.0
            tcv2(k) = 0.0
            tcv3(k) = 0.0
         enddo
*
      endif
*
      do i=1,10
         pent(i)=0.0
      enddo
       do i=1,50
         hexa(i)=0.0
         hexb(i)=0.0
         hexc(i)=0.0
       enddo
       do i=1,40
          hexd(i)=0.0
       enddo
       do i=1,20
          hdh(i)=0.0
          veto(i)=0.0
       enddo
*
       do k=1,20
```

```
         pv(k) = 0.0
      enddo
*
      do k=1,5
         cv(k) = 0.0
      enddo
*
      etgt=0.0
      etarg=0.0
      epveto=0.0
      eclad=0.0
*
      END
*
*CMZ :          26/05/89  08.50.35  by  Rene Brun
*-- Author :
      SUBROUTINE GUPHAD
*
*    ******************************************************************
*    *                                                                *
*    *                                                                *
*    *       GEANT3 user routine called at each step                  *
*    *       to evaluate the next hadronic interaction point          *
*    *                                                                *
*    *                (TATINA) uses R.Barlow cross-sections tables    *
*    *                (GHEISHA)uses the GHEISHA cross-sections        *
*    *                                                                *
*    ******************************************************************
*
*    ------------------------------------------------------------------
*
      CALL GPGHEI
*
      END
*CMZ :          26/05/89  08.50.35  by  Rene Brun
*-- Author :
      SUBROUTINE GUHADR
*
*    ******************************************************************
*    *                                                                *
*    *                                                                *
*    *       GEANT3 user routine called when a hadronic process       *
*    *       has been selected in the current step.                   *
*    ******************************************************************
*
*    ------------------------------------------------------------------
*
      CALL GHEISH
*
```

```
      END
*CMZ :          13/05/89  19.04.22  by  Rene Brun
*-- Author :
      SUBROUTINE UGLAST
*
************************************************************************
*                                                                      *
*           Termination routine to print histograms and statistics    *
*                                                                      *
************************************************************************
*
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
*
*     -----------------------------------------------------------------
*
      CALL GLAST
*
      CALL HROUT(0,ICYCLE,' ')
      CALL HREND('PIBETA')
      CLOSE(20)
      CALL IGEND
*
      END
*CMZ :          13/05/89  19.04.22  by  Rene Brun
*-- Author :

      SUBROUTINE VIEWYZ(IVIEW)
*
        call gsatt('MAMA','SEEN',0)
        call gsatt('CALO','SEEN',0)
      END

      SUBROUTINE QNEXT

      END
*
      SUBROUTINE DECAY3B(WMC,ME_SQ)
*
      IMPLICIT REAL*4 (B-H,M,O-Z)
      COMMON/GENIN/NP,ATECM,AMASS(18),KGENEV
      COMMON/MAWUSE/E1,X1,Y1,Z1,E2,X2,Y2,Z2,E3,X3,Y3,Z3
      COMMON/GENOUT/APCM(5,18),AWT
*
      VOLATILE/MAWUSE/
*
      NP=3
      MP0=0.13497340E0
      MEP=0.000510999E0
```

```
      MNU=0.0E0
      AMASS(1)=MP0
      AMASS(2)=MEP
      AMASS(3)=MNU
      KGENEV=1
      ATECM=0.139567550
      M0=0.139567550E0
*
      CALL GENBOD
*
      X1= APCM(1,1)
      Y1= APCM(2,1)
      Z1= APCM(3,1)
      E1= APCM(4,1)
*
      X2= APCM(1,2)
      Y2= APCM(2,2)
      Z2= APCM(3,2)
      E2= APCM(4,2)
*
      X3= APCM(1,3)
      Y3= APCM(2,3)
      Z3= APCM(3,3)
      E3= APCM(4,3)
*
      WMC = AWT
*
      ME_SQ = ABS(2.0*(-M0*E2+X1*X2+Y1*Y2+Z1*Z2-E1*E2)*
     1              (-M0*E3+X1*X3+Y1*Y3+Z1*Z3-E1*E3)-
     1              (X1*X1+Y1*Y1+Z1*Z1-(M0+E1)*(M0+E1))*
     1              (X2*X3+Y2*Y3+Z2*Z3-E2*E3))
*
      return
      END
*

      SUBROUTINE DECAY_PI0
*
      IMPLICIT REAL*4 (e,p,t,r,b,g,m)
      INTEGER ik,iket
*
      COMMON/ketevi/thetal,phil,pgam(8),thetag(2),phig(2)
     1             ,plpi0,elpi0,relang,beta,gamma
      dimension gam(8)
      logical check1,check2
      VOLATILE/ketevi/
      DATA MP0/0.134973400/
      DATA ISEED/96835/
*     generate random CM angles thetac and phic:
```

```
 10   continue
      phic = 360. * RAN(ISEED)
      thetac = acosd(1. - 2 * RAN(ISEED))
*     set the kinematics parameters:
      beta = plpi0/elpi0
      gamma = 1 / SQRT(1 - beta*beta)
      iket = 1
      ik   = 0
      do while (iket.le.2)
         pgx = (MP0/2.)*sind(thetac)*cosd(phic)
         pgy = (MP0/2.)*sind(thetac)*sind(phic)
         pgz = (MP0/2.)*gamma*(cosd(thetac) + beta)
         eg  = (MP0/2.)*gamma*(1 + beta*cosd(thetac))
         gam(ik+iket) = pgz*sind(thetal)*cosd(phil) +
     1               pgx*cosd(thetal)*cosd(phil) -
     1               pgy*sind(phil)
         gam(iket+ik+1) = pgz*sind(thetal)*sind(phil) +
     1                 pgx*cosd(thetal)*sind(phil) +
     1                 pgy*cosd(phil)
         gam(iket+2+ik) = pgz*cosd(thetal) - pgx*sind(thetal)
         gam(iket+3+ik) = eg
         iket = iket + 1
         thetac = - thetac + 180.
         phic = 180. + phic
         ik = ik + 3
      end do
      pgx1 = gam(1)
      pgy1 = gam(2)
      pgz1 = gam(3)
      pgx2 = gam(5)
      pgy2 = gam(6)
      pgz2 = gam(7)
      thetag(1) = 57.29578*acos(pgz1/sqrt(pgx1*pgx1+
     1                        pgy1*pgy1+pgz1*pgz1))
      phig(1) = 57.29578*atan2(pgy1,pgx1)

      thetag(2) = 57.29578*acos(pgz2/sqrt(pgx2*pgx2+
     1                        pgy2*pgy2+pgz2*pgz2))
      phig(2) = 57.29578*atan2(pgy2,pgx2)
      relang = 57.29578*acos((pgx1*pgx2+pgy1*pgy2+pgz1*pgz2)
     1            /(sqrt(pgx1*pgx1+pgy1*pgy1+pgz1*pgz1)*
     1              sqrt(pgx2*pgx2+pgy2*pgy2+pgz2*pgz2)))
*
      do i=1,8
        pgam(i)=gam(i)
      enddo
      return
      END
*
```

```
      FUNCTION gasdev(idum)
      INTEGER idum
      REAL gasdev
      INTEGER iset
      REAL fac,gset,rsq,v1,v2,dum
      SAVE iset,gset
      DATA iset/0/
      if (iset.eq.0) then
        dum = real(idum)
1       v1=2.*RNDM(dum)-1.
        dum = sqrt(20.0)*real(idum)
        v2=2.*RNDM(dum)-1.
        rsq=v1**2+v2**2
        if(rsq.ge.1..or.rsq.eq.0.)goto 1
        fac=sqrt(-2.*log(rsq)/rsq)
        gset=v1*fac
        gasdev=v2*fac
        iset=1
      else
        gasdev=gset
        iset=0
      endif
      return
      END
*
      FUNCTION poidev(xm,idum)
      INTEGER idum
      REAL poidev,xm,PI
      PARAMETER (PI=3.141592654)
      REAL alxm,em,g,oldm,sq,t,y,gammln,ran
      SAVE alxm,g,oldm,sq
      DATA oldm /-1./
      if (xm.lt.12.)then
        if (xm.ne.oldm) then
          oldm=xm
          g=exp(-xm)
        endif
        em=-1
        t=1.
2       em=em+1.
        t=t*ran(idum)
        if (t.gt.g) goto 2
      else
        if (xm.ne.oldm) then
          oldm=xm
          sq=sqrt(2.*xm)
          alxm=log(xm)
          g=xm*alxm-gammln(xm+1.)
        endif
```

```
1       y=tan(PI*ran(idum))
        em=sq*y+xm
        if (em.lt.0.) goto 1
        em=int(em)
        t=0.9*(1.+y**2)*exp(em*alxm-gammln(em+1.)-g)
        if (ran(idum).gt.t) goto 1
      endif
      poidev=em
      return
      END

*
      FUNCTION gammln(xx)
      REAL gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     *24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     *-.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
      do 11 j=1,6
        y=y+1.d0
        ser=ser+cof(j)/y
11    continue
      gammln=tmp+log(stp*ser/x)
      return
      END
        SUBROUTINE MAT_EL(T1,P1,T2,P2,T3,P3)
*
        COMMON/ELEMENTS/T1P,P1P,T2P,P2P,T3P,P3P
*
        a11 = sind(t3p)*cosd(p3p)
        a12 = sind(t3p)*sind(p3p)
        a13 = cosd(t3p)
*
        a31 = -sind(p3p)
        a32 = cosd(p3p)
        a33 = 0.0
*
        a21 = cosd(p3p)*cosd(t3p)
        a22 = cosd(t3p)*sind(p3p)
        a23 = -sind(t3p)
*
        x1 = sind(t1)*cosd(p1)
```

```
       x2 = sind(t1)*sind(p1)
       x3 = cosd(t1)
*
       y1 = sind(t2)*cosd(p2)
       y2 = sind(t2)*sind(p2)
       y3 = cosd(t2)
*
       z1 = sind(t3)*cosd(p3)
       z2 = sind(t3)*sind(p3)
       z3 = cosd(t3)
*
       xp1 = a11*x1+a12*x2+a13*x3
       xp2 = a21*x1+a22*x2+a23*x3
       xp3 = a31*x1+a32*x2+a33*x3
*
       yp1 = a11*y1+a12*y2+a13*y3
       yp2 = a21*y1+a22*y2+a23*y3
       yp3 = a31*y1+a32*y2+a33*y3
*
       zp1 = a11*z1+a12*z2+a13*z3
       zp2 = a21*z1+a22*z2+a23*z3
       zp3 = a31*z1+a32*z2+a33*z3
*
       t1 = acosd(xp3)
       p1 = atan2d(xp2,xp1)
       if (p1.lt.0) p1=p1+360.0
*
       t2 = acosd(yp3)
       p2 = atan2d(yp2,yp1)
       if (p2.lt.0) p2=p2+360.0
*
       t3 = acosd(zp3)
       p3 = atan2d(zp2,zp1)
       if (p3.lt.0) p3=p3+360.0
*
       RETURN
       END
C    +PATCH,W515.
C     +DECK,GENBOD.

      SUBROUTINE GENBOD

C   SUBROUTINE TO GENERATE N-BODY EVENT
C   ACCORDING TO FERMI LORENTZ-INVARIANT PHASE SPACE
C   ADAPTED FROM FOWL (CERN W505) SEPT. 1974 BY F. JAMES
C   EVENTS ARE GENERATED IN THEIR OWN CENTER-OF-MASS,
C   BUT MAY BE TRANSFORMED TO ANY FRAME USING LOREN4
C
C   INPUT TO SUBROUTINE IS THRU COMMON BLOCK GENIN
```

```
C            NP=NUMBER OF OUTGOING PARTICLES (.LT. 19)
C            TECM=TOTAL ENERGY IN CENTER-OF-MASS
C            AMASS(I)=MASS OF ITH OUTGOING PARTICLE
C            KGENEV=1 FOR CONSTANT CROSS SECTION
C                    2 FOR FERMI ENERGY-DEPENDANCE
C
C   OUTPUT FROM SUBROUTINE IS THRU COMMON BLOCK GENOUT
C            PCM(1,I)=X-MOMENTUM IF ITH PARTICLE
C            PCM(2,I)=Y-MOMENTUM IF ITH PARTICLE
C            PCM(3,I)=Z-MOMENTUM IF ITH PARTICLE
C            PCM(4,I)=ENERGY OF ITH PARTICLE
C            PCM(5,I)=MOMENTUM OF ITH PARTICLE
C            WT=WEIGHT OF EVENT

      COMMON/GENIN / NP, TECM, AMASS(18), KGENEV
      COMMON/GENOUT/ PCM(5,18) , WT

      DIMENSION EMM(18)
      DIMENSION RNO(50)

C--       PCM1 IS LINEAR EQUIV. OF PCM TO AVOID DOUBLE INDICES

      DIMENSION EM(18),PD(18),EMS(18),SM(18),FFQ(18),PCM1(90)
      EQUIVALENCE (NT,NP),(AMASS(1),EM(1)),(PCM1(1),PCM(1,1))

C FFQ(N)=PI * (TWOPI)**(N-2) / (N-2)FACTORIAL

      DATA FFQ/0.,3.141592, 19.73921, 62.01255, 129.8788, 204.0131,
     2     256.3704, 268.4705, 240.9780, 189.2637,
     3     132.1308,  83.0202,  47.4210,  24.8295,
     4     12.0006,   5.3858,   2.2560,   0.8859/

      DATA KNT,TWOPI/0,6.2831853073/

C    +SELF,IF=CDC.

      DATA KGENEV/ 2 /

C    +SELF.

C        INITIALIZATION
      KNT=KNT + 1

      IF(KNT.GT.1) GOTO 100
        WRITE(6,1160)
        WRITE(6,1200) NP,TECM,(AMASS(JK),JK=1,NP)
 100   CONTINUE

      IF(NT.LT.2) GOTO 1001
```

```
      IF(NT.GT.18) GOTO 1002

      NTM1=NT-1
      NTM2=NT-2
      NTP1=NT+1
      NTNM4=3*NT - 4
      EMM(1)=EM(1)
      TM=0.0
      DO 200 I=1,NT
      EMS(I)=EM(I)**2
      TM=TM+EM(I)
 200    SM(I)=TM

C         CONSTANTS DEPENDING ON TECM

      TECMTM=TECM-TM
      IF(TECMTM.LE.0.0) GOTO 1000
      EMM(NT)=TECM

      IF(KGENEV.GT.1) GOTO 400
C         CONSTANT CROSS SECTION AS FUNCTION OF TECM
      EMMAX=TECMTM+EM(1)
      EMMIN=0.0
      WTMAX=1.0
      DO 350 I=2,NT
      EMMIN=EMMIN+EM(I-1)
      EMMAX=EMMAX+EM(I)
 350   WTMAX=WTMAX*PDK(EMMAX,EMMIN,EM(I))
      WTMAXQ=1.0/WTMAX

      GOTO 455

C--       FERMI ENERGY DEPENDENCE FOR CROSS SECTION

 400   WTMAXQ=TECMTM**NTM2*FFQ(NT) / TECM

C         CALCULATION OF WT BASED ON EFFECTIVE MASSES EMM
 455    CONTINUE

C--                FILL RNO WITH 3*NT-4 RANDOM NUMBERS,
C--             OF WHICH THE FIRST NT-2 ARE ORDERED.

      DO 457 I= 1, NTNM4
 457      RNO(I)=RNDM(I)

      IF(NTM2) 900,509,460
 460   CONTINUE
```

```
      CALL FLPSOR(RNO,NTM2)

      DO 508 J=2,NTM1

 508      EMM(J)=RNO(J-1)*(TECMTM)+SM(J)
 509       WT=WTMAXQ
      IR=NTM2
      DO 530 I=1,NTM1
      PD(I)=PDK(EMM(I+1),EMM(I),EM(I+1))

 530   WT=WT*PD(I)

C--       COMPLETE SPECIFICATION OF EVENT (RAUBOLD-LYNCH METHOD)

      PCM(1,1)=0.0
      PCM(2,1)=PD(1)
      PCM(3,1)=0.0
      DO 570 I=2,NT
      PCM(1,I)=0.0
      PCM(2,I)=-PD(I-1)
      PCM(3,I)=0.0
      IR=IR+1
      BANG=TWOPI*RNO(IR)
      CB=COS(BANG)
      SB=SIN(BANG)
      IR=IR+1
      C=2.0*RNO(IR)-1.0
      S=SQRT(1.0-C*C)
      IF(I.EQ.NT) GOTO 1567
      ESYS=SQRT(PD(I)**2+EMM(I)**2)
      BETA=PD(I)/ESYS
      GAMA=ESYS/EMM(I)
      DO 568 J=1,I
      NDX=5*J - 5
      AA= PCM1(NDX+1)**2 + PCM1(NDX+2)**2 + PCM1(NDX+3)**2
      PCM1(NDX+5)=SQRT(AA)
      PCM1(NDX+4)=SQRT(AA+EMS(J))

      CALL ROTES2(C,S,CB,SB,PCM,J)

      PSAVE=GAMA*(PCM(2,J)+BETA*PCM(4,J))
 568   PCM(2,J)=PSAVE
      GOTO 570

 1567  DO 1568 J=1,I
      AA=PCM(1,J)**2 + PCM(2,J)**2 + PCM(3,J)**2
      PCM(5,J)=SQRT(AA)
      PCM(4,J)=SQRT(AA+EMS(J))
```

```
      CALL ROTES2(C,S,CB,SB,PCM,J)

 1568  CONTINUE
 570    CONTINUE
 900     CONTINUE

      RETURN

C          ERROR RETURNS
 1000  WRITE(6,1100)
      GOTO 1050
 1001  WRITE(6,1101)
      GOTO 1050
 1002  WRITE(6,1102)
 1050   WRITE(6,1150) KNT
      WRITE(6,1200) NP,TECM,(AMASS(JK),JK=1,NP)

      STOP

 1100  FORMAT(28H0 AVAILABLE ENERGY NEGATIVE )
 1101   FORMAT(33H0 LESS THAN 2 OUTGOING PARTICLES )
 1102    FORMAT(34H0 MORE THAN 18 OUTGOING PARTICLES )
 1150     FORMAT(47H0 ABOVE ERROR DETECTED IN GENBOD AT CALL NUMBER,I7)
 1160      FORMAT(34H0 FIRST CALL TO SUBROUTINE GENBOD )
 1200       FORMAT(36H  INPUT DATA TO GENBOD.        NP=   ,I6/
     +  ,8H   TECM=,E16.7,18H  PARTICLE MASSES=,5E15.5/(42X,5E15.5)
     +)

      END

C ---------------------------------------------------------------------------
C     +DECK,PDK.

      FUNCTION PDK(A,B,C)

C     CALLED FROM -  GENBOD
C     PDK = SQRT(A*A+(B*B-C*C)**2/(A*A) - 2.0*(B*B+C*C))/2.0

      REAL*8 A_D,  B_D,  C_D,
     +       A2_D, B2_D, C2_D,
     +       TEST, PDK_D

      A_D = DBLE(A)
      B_D = DBLE(B)
      C_D = DBLE(C)

      A2_D = A_D*A_D
      B2_D = B_D*B_D
      C2_D = C_D*C_D
```

```
      TEST = A2_D + (B2_D-C2_D)**2/A2_D - 2.0*(B2_D+C2_D)

      IF(TEST.LT.0.) THEN
        WRITE(6,*) 'TEST is LT zero', TEST
        PDK_D = 0.0
      ELSE
        PDK_D = 0.5 * DSQRT(TEST)
      ENDIF

      PDK = REAL(PDK_D)

      RETURN
      END
```