```
C...As an example, consider a main program of the form
C...Double precision and integer declarations.
*======================================================================
      program pyMainERHIC

      include 'pythia.inc'              ! All PYTHIA commons blocks
      include "mc_set.inc"
      include "py6strf.inc"
      include "mcRadCor.inc"
      include "radgen.inc"
      include "phiout.inc"

      integer NEV, NPRT, ievent, genevent, I, tracknr, ltype
      integer lastgenevent, idum1, idum2, initseed, nrtrack
      REAL trueX, trueW2, trueNu
      DOUBLE PRECISION sqrts, radgamE, radgamp, radgamEnucl
      DOUBLE PRECISION pbeamE, pbeta, pgamma, ebeamE, epznucl
      CHARACTER PARAM*100
      LOGICAL UseLut, GenLut

c ---------------------------------------------------------------------
c     Run parameter
c ---------------------------------------------------------------------
      integer*4 today(3), now(3)
c---------------------------------------------------------------------
c     ASCII output file
c ---------------------------------------------------------------------
      integer asciiLun
      parameter (asciiLun=29)
      CHARACTER*256 outputfilename
      CHARACTER*256 outname

c---------------------------------------------------------------------
! ... force block data modules to be read
C       external pydata
c ---------------------------------------------------------------------

       iModel=0
       pbeam=100.
       ebeam=4.0
       etype=11
       masse=PYMASS(11)
       massp=PYMASS(2212)
       ievent=0
       genevent=0
       lastgenevent=0
       tracknr=0

C...Read output file name
       READ(*,*) outname
C...Read lepton beam type
       READ(*,*) ltype
C...Read parameters for PYINIT call (beam and target particle energy).
       READ(*,*) pbeam, ebeam
C...Read number of events to generate, and to print.
       READ(*,*) NEV,NPRT
C...Read min/max x of radgen lookup table
       READ(*,*) mcSet_XMin, mcSet_XMax
C...Read min/max y of generation range
       READ(*,*) mcSet_YMin, mcSet_YMax
C...Read min/max Q2 of generation range
       READ(*,*) mcSet_Q2Min, mcSet_Q2Max
C...Read information for cross section used in radgen
       READ(*,*) genSet_FStruct, genSet_R
C...Read parameters of radcorr: do radcorr (1), generate look-up table (2)
       READ(*,*) qedrad
C...Read parameters for PYTHIA-Model = which generation is done
       READ(*,*) iModel
C...Read target type mass and charge
       READ(*,*) mcSet_TarA, mcSet_TarZ
C...Read information for cross section used in radgen
  100  READ(*,'(A)',END=200) PARAM
       CALL PYGIVE(PARAM)
       GOTO 100
c ---------------------------------------------------------------------
C...Initialize PYTHIA.
c ---------------------------------------------------------------------
  200  write(*,*) '*********************************************'
       write(*,*) 'NOW all parameters are read by PYTHIA'
       write(*,*) '*********************************************'
C       call PYLIST(11)
C       call PYLIST(12)

C     Getting the date and time of the event generation

      call idate(today)   ! today(1)=day, (2)=month, (3)=year
      call itime(now)     ! now(1)=hour, (2)=minute, (3)=second

!     Take date as the SEED for the random number generation

      initseed = today(1) + 10*today(2) + today(3) + now(1) + 5*now(3)
      write(6,*) 'SEED = ', initseed
      call rndmq (idum1,idum2,initseed,' ')

      sqrts=sqrt(4*pbeam*ebeam)
      write(*,*) '**********************************************'
      write(*,*) 'proton beam energy:', pbeam, 'GeV'
      write(*,*) 'lepton beam energy:', ebeam, 'GeV'
      write(*,*) 'resulting sqrt(s):', sqrts, 'GeV'
      write(*,*) '**********************************************'
C     proton is defined in positive z and as target
      P(2,1)=0.0
      P(2,2)=0.0
      P(2,3)=pbeam
C     lepton is defined in negative z and as beam
      P(1,1)=0.0
      P(1,2)=0.0
      P(1,3)=-ebeam

      if (mcSet_TarZ.eq.0) then
        massp=PYMASS(2112)
      else
        massp=PYMASS(2212)
      endif
      masse=PYMASS(ltype)

      pbeamE=sqrt(pbeam**2+massp**2)
      pbeta=pbeam/pbeamE
      pgamma=pbeamE/massp
      ebeamE=sqrt(ebeam**2+masse**2)
      ebeamEnucl=pgamma*ebeamE-pgamma*pbeta*(-ebeam)
      epznucl=-pgamma*pbeta*(ebeamE)+pgamma*(-ebeam)
      write(*,*) ebeamEnucl, ebeamE, epznucl, -ebeam
      mcSet_EneBeam=sngl(ebeamEnucl)

       if (iModel.eq.0) then
           UseLUT=.false.
           GenLUT=.false.
           qedrad=0
           MSTP(199)=0
           mcRadCor_EBrems=0.
       elseif (iModel.eq.1) then
         if (qedrad.eq.0) then
             mcRadCor_EBrems=0.
             UseLUT=.false.
             GenLUT=.false.
             MSTP(199)=1
         elseif (qedrad.eq.1) then
```

```
             mcRadCor_EBrems=0.
             UseLUT=.true.
             GenLUT=.false.
             MSTP(199)=1
             call radgen_init(UseLUT,GenLUT)
             write(*,*) 'I have initialized radgen'
         elseif (qedrad.eq.2) then
             write(*,*) 'radgen lookup table will be generated'
             mcRadCor_EBrems=0.
             UseLUT=.true.
             GenLUT=.true.
             MSTP(199)=1
             call radgen_init(UseLUT,GenLUT)
             goto 500
         endif
       endif

      call pyinit ('3MOM','gamma/e-','p+',WIN)

C      If we ever want to simulate fixed target we need to change this
C      win=ebeam
C      call pyinit('fixt','gamma/e-','p+', WIN)

c ---------------------------------------------------------------------
c     Open ascii output file
c ---------------------------------------------------------------------
       outputfilename=outname
       open(asciiLun, file=outputfilename)
       write(*,*) 'the outputfile will be named: ', outname

c ---------------------------------------------------------------------
C...Event generation loop
c ---------------------------------------------------------------------

C   This is what we write in the ascii-file

        write(29,*)' PYTHIA EVENT FILE '
        write(29,*)'============================================'
        write(29,30)
30      format('I, ievent, genevent, subprocess, nucleon,
     &  targetparton, xtargparton, beamparton, xbeamparton,
     &  thetabeamprtn, truey, trueQ2, truex, trueW2, trueNu, leptonphi,
     &  s_hat, t_hat, u_hat, pt2_hat, Q2_hat, F2, F1, R, sigma_rad,
     &  SigRadCor, EBrems, photonflux, nrTracks')
        write(29,*)'============================================'
        write(29,*)' I  K(I,1)  K(I,2)  K(I,3)  K(I,4)  K(I,5)
     &  P(I,1)  P(I,2)  P(I,3)  P(I,4)  P(I,5)  V(I,1)  V9I,2)  V(I,3)'
        write(29,*)'============================================'

       DO 300 IEV=1,NEV
999      CALL PYEVNT
         if (MSTI(61).eq.1) then
            write(*,*) 'go back to PYEVNT call'
            goto 999
         endif
C         CALL PYLIST(2)

         ievent=IEV
         genevent=NGEN(0,3)-lastgenevent

       trueX =  VINT(307)/VINT(309)/(4*pbeam*ebeam)
       trueW2 = massp**2 + VINT(307)*(1/trueX-1)
       trueNu = (trueW2 + VINT(307) - massp**2)/(2.*massp)
      if (mcRadCor_EBrems.gt.0.) then
         radgamEnucl=sqrt(dplabg(1)**2+dplabg(2)**2+dplabg(3)**2)
         radgamE=pgamma*radgamEnucl-pgamma*pbeta*dplabg(3)
         radgamp=-pgamma*pbeta*radgamEnucl+pgamma*dplabg(3)
C         write(*,*) radgamEnucl, radgamE, dplabg(3), radgamp
      else
        radgamEnucl=0D0
        radgamE=0D0
        radgamp=0D0
      endif

         tracknr=N
         if (mcRadCor_EBrems.gt.0.) then
            nrtrack=tracknr+1
         else
            nrtrack=tracknr
         endif

         if ((msti(1).ge.91).and.(msti(1).le.94)) msti(16)=0

         write(29,32) 0, ievent, genevent, msti(1), msti(12),
     &        msti(16), pari(34), msti(15), pari(33), pari(53),
     &        VINT(309), VINT(307), trueX, trueW2, trueNu,
     &        VINT(313), pari(14), pari(15), pari(16),
     &        pari(18),  pari(22), sngl(py6f2), sngl(py6f1),
     &        py6r, mcRadCor_Sigrad, mcRadCor_sigcor, radgamEnucl,
     &        VINT(319), nrtrack
 32      format((I4,1x,$),(I10,1x,$),3(I4,1x,$),(I10,1x,$),f9.6,1x,$,
     &         I12,1x,$,
     &         2(f12.6,1x,$),7(f18.11,3x,$),11(f19.9,3x,$),I12,/)
         write(29,*)'============================================'

         DO I=1,tracknr
         if (K(I,3).le.nrtrack) then
         write(29,34) I,K(I,1),K(I,2),K(I,3),K(I,4),K(I,5),
     &        P(I,1),P(I,2),P(I,3),P(I,4),P(I,5),
     &        V(I,1),V(I,2),V(I,3)
         endif
         ENDDO
         if (mcRadCor_EBrems.gt.0.) then
            write(29,34) nrtrack, 55, 22, 1, 0, 0,
     &      sngl(dplabg(1)),sngl(dplabg(2)),sngl(-radgamp),
     &      sngl(radgamE), 0., 0., 0., 0.
         endif
 34      format(2(I6,1x,$),I10,1x,$,3(I8,1x,$),8(f15.6,1x,$),/)
         write(29,*)'=============== Event finished ==============='
         lastgenevent=NGEN(0,3)

  300  CONTINUE

C...Print cross sections.
       CALL PYSTAT(1)
       CALL PYSTAT(4)

       write(*,*)"The charm mass used is: ", PMAS(4,1)

C...Print the Pythia cross section which is needed to get an absolut
C   normalisation the number is in microbarns
       write(*,*)'==================================================='
       write(*,*)'Pythia total cross section normalisation:',
     &           pari(1)*1000, ' microbarn'
       write(*,*)'Total Number of generated events', MSTI(5)
       write(*,*)'Total Number of trials', NGEN(0,3)
       write(*,*)'==================================================='
       close(29)

  500  if (qedrad.eq.2) then
         write(*,*) 'lookup table is generated;'
         write(*,*) 'to run now pythia change parameter qedrad to 1'
       endif

       END
```