```
C
      PROGRAM SMERG
C
C Fortran-77 sort-merge solution for Siena exercise
C
      PARAMETER(NX=2000000)
      INTEGER IH(NX),IK(NX),IL(NX),IP(NX),IQ(NX)
      REAL FF(NX),SI(NX),SY(12,24)
C
C Read data from standard input
C
      READ(*,'(/)')
      READ(*,*)NS
      READ(*,*)((SY(I,J),I=1,12),J=1,NS)
      NR=0
  1   N=NR+1
      IF(N.GT.NX)STOP '** Too many reflections **'
      READ(*,*,END=5)IH(N),IK(N),IL(N),FF(N),SI(N)
      IP(N)=N
      NR=N
C
C Convert reflection indices to standard setting
C
      U=REAL(IH(N))
      V=REAL(IK(N))
      W=REAL(IL(N))
        DO 4 M=-1,1,2
          DO 3 J=1,NS
          I=M*NINT(SY(1,J)*U+SY(4,J)*V+SY(7,J)*W)
          K=M*NINT(SY(2,J)*U+SY(5,J)*V+SY(8,J)*W)
          L=M*NINT(SY(3,J)*U+SY(6,J)*V+SY(9,J)*W)
          IF(L.LT.IL(N))GOTO 3
          IF(L.GT.IL(N))GOTO 2
          IF(K.LT.IK(N))GOTO 3
          IF(K.GT.IK(N))GOTO 2
          IF(I.LE.IH(N))GOTO 3
  2       IH(N)=I
          IK(N)=K
          IL(N)=L
  3       CONTINUE
  4     CONTINUE
      GOTO 1
C
C Sort pointer arrays on h, then k, then l
C
  5   CALL INSORT(NR,IP,IQ,IH)
      CALL INSORT(NR,IQ,IP,IK)
      CALL INSORT(NR,IP,IQ,IL)
C
C Combine equivalents - now contiguous
C
      R=0.
      S=0.
      NU=NR
      NC=0
      N=1
  6   IF(N.GT.NR)GOTO 16
      M=N
      MQ=IQ(M)
      P=0.
      Q=0.
  7   NQ=IQ(N)
      IF(IH(NQ).NE.IH(MQ).OR.IK(NQ).NE.IK(MQ).OR.IL(NQ).NE.IL(MQ))GOTO 8
      P=P+FF(NQ)
      Q=Q+1./SI(NQ)**2
      N=N+1
      IF(N.LE.NR)GOTO 7
```

```
  8    P=P/REAL(N-M)
       Q=1./SQRT(Q)
C
C Detect systematic absences and centric reflections
C
       U=REAL(IH(MQ))
       V=REAL(IK(MQ))
       W=REAL(IL(MQ))
       NT=0
         DO 13 J=2,NS
         I=NINT(SY(1,J)*U+SY(4,J)*V+SY(7,J)*W)
         K=NINT(SY(2,J)*U+SY(5,J)*V+SY(8,J)*W)
         L=NINT(SY(3,J)*U+SY(6,J)*V+SY(9,J)*W)
         IF(I.NE.IH(MQ).OR.K.NE.IK(MQ).OR.L.NE.IL(MQ))GOTO 12
         IF(MOD(NINT(12.*(SY(10,J)*U+SY(11,J)*V+SY(12,J)*W)),12).EQ.0)
     +   GOTO 13
 11      FORMAT(' Reflection',3i4,'  syst. abs.,  I/sigma =',F8.2)
         WRITE(*,11)IH(MQ),IK(MQ),IL(MQ),P/Q
         GOTO 6
 12      IF(-I.EQ.IH(MQ).AND.-K.EQ.IK(MQ).AND.-L.EQ.IL(MQ))NT=1
 13      CONTINUE
       NC=NC+NT
       NU=NU+1
       IF(NU.GT.NX)STOP '** Too many reflections **'
       IH(NU)=IH(MQ)
       IK(NU)=IK(MQ)
       IL(NU)=IL(MQ)
       FF(NU)=P
       SI(NU)=Q
       IF(N.LE.M+1)GOTO 6
         DO 14 I=M,N-1
         R=R+ABS(FF(IQ(I))-P)
         S=S+P
 14      CONTINUE
       GOTO 6
C
C Unique reflections now in IH(NR+1..NU) etc. Output statistics.
C
 15    FORMAT(/I6,' Unique reflections, of which',I5,' centric   ',
     +'R(int) =',F8.4/)
 16    WRITE(*,15)NU-NR,NC,R/S
       END
C
C ---------------------------------------------------------------------
C
       SUBROUTINE INSORT(N,IP,IQ,ID)
C
C Sort-merge integer data in order of ascending ID(I). IP is the
C current pointer array to ID and IQ becomes the new pointer array.
C
       INTEGER IP(N),IQ(N),ID(N),IT(9999)
       L=ID(1)
       M=L
         DO 1 I=2,N
         L=MIN0(ID(I),L)
         M=MAX0(ID(I),M)
   1     CONTINUE
       L=L-1
       M=M-L
         DO 2 I=1,M
         IT(I)=0
   2     CONTINUE
         DO 3 I=1,N
         J=ID(I)-L
         IT(J)=IT(J)+1
   3     CONTINUE
       J=0
```

```
      DO 4 I=1,M
      K=J
      J=J+IT(I)
      IT(I)=K
4     CONTINUE
      DO 5 I=1,N
      J=ID(IP(I))-L
      IT(J)=IT(J)+1
      J=IT(J)
      IQ(J)=IP(I)
5     CONTINUE
   RETURN
   END
```