```
      subroutine errorf(x,erf,erfc,expnon)
c input: x
c output: erf, erfc, expnon
      implicit double precision (a-h,o-z)
c ************************************************************************
c this subroutine is based on an Algol 60 program from the
c Numal library (section 6.7).
c the used method is based on the paper
c      W.J. Cody, 1969,
c      Rational Chebyshev Approximations for the Error Function,
c      Mathematics of Computation, vol. 23, pp. 631-637
c conversion for the VAX 11/750 by E.J.M. Veling, 11 October 1984.
c ************************************************************************
c errorf calculates in erf:  erf(x),
c                  in erfc: erfc(x),
c                  in expnon: x>=0, exp(x*x)*erfc(x),
c                             x< 0, exp(x*x)*(2-erfc(x)).
c errorf needs double precision input and gives double precision output.
c this subroutine calculates the errorfunctions with a relative precision
c of the order of magnitude of 10**(-12).
c the fact that the relative precision is so small makes this subroutine
c useful at the evaluation of analytic expressions.
c if x >  9.1,  erf =  1, erfc = 0,
c if x < -5.5,  erf = -1, erfc = 2,
c to prevent under- and overflow; in these cases the variable expnon can
c be used.
c ************************************************************************
      data c1,c2,c3,c4,c5,c6,c7/
     1   -0.0356098437018154d0,
     2    6.99638348861914d0,
     3   21.9792616182942d0,
     4  242.667955230532d0,
     5   15.0827976304078d0,
     6   91.1649054045149d0,
     7  215.058875869861d0/
      data d1,d2,d3,d4,d5,d6,d7,d8/
     1    -0.00000013684857382717d0,
     2     0.564195517478974d0,
     3     7.21175825088309d0,
     4    43.1622272220567d0,
     5   152.989285046940d0,
     6   339.320816734344d0,
     7   451.918953711873d0,
     8   300.459261020162d0/
      data e1,e2,e3,e4,e5,e6,e7/
     1    12.7827273196294d0,
     2    77.0001529352295d0,
     3   277.585444743988d0,
     4   638.980264465631d0,
     5   931.354094850610d0,
     6   790.950925327898d0,
     7   300.459260956983d0/
      data f1,f2,f3,f4,f5/
     1     0.0223192459734185d0,
     2     0.278661308609648d0,
     3     0.226956593539687d0,
     4     0.0494730910623251d0,
```

```
     5     0.00299610707703542d0/
      data g1,g2,g3,g4,g5/
     1     1.98733201817135d0,
     2     1.05167510706793d0,
     3     0.191308926107830d0,
     4     0.0106209230528468d0,
     5     0.564189583547756d0/
400   absx=dabs(x)
      if (absx.le.0.5d0) goto 500
      goto 600
500   c=x*x
      p=((c1*c+c2)*c+c3)*c+c4
      q=((c+c5)*c+c6)*c+c7
      erf=x*p/q
      erfc=1.d0-erf
      expnon=dexp(x*x)*(1.d0-dsign(1.d0,x)*erf)
      return
600   if (absx.lt.4.d0) goto 700
      goto 800
700   c=abs(x)
      p=((((((d1*c+d2)*c+d3)*c+d4)*c+d5)*c+d6)*c+d7)*c+d8
      q=((((((c+e1)*c+e2)*c+e3)*c+e4)*c+e5)*c+e6)*c+e7
      expnon=p/q
      if (x.gt.0.d0) goto 710
      goto 720
710   erfc=p/q*dexp(-x*x)
      erf=1.d0-erfc
      return
720   erfc=2.d0-p/q*dexp(-x*x)
      erf=1.d0-erfc
      return
800   c=1.d0/x/x
      p=(((f1*c+f2)*c+f3)*c+f4)*c+f5
      q=(((c+g1)*c+g2)*c+g3)*c+g4
      c=(c*(-p)/q+g5)/absx
      expnon=c
      if ((x.gt.0.d0).and.(x.le.9.1d0)) goto 810
      goto 820
810   erfc=c*dexp(-x*x)
      erf=1.d0-erfc
      return
820   if ((x.lt.0.d0).and.(x.ge.-5.5d0)) then
         erfc=2.d0-c*dexp(-x*x)
         erf=1.d0-erfc
         return
      endif
      if (x.gt.0.d0) then
         erf=1.d0
         erfc=0.d0
         return
      else
         erf=-1.d0
         erfc=2.d0
         return
      endif
      end
c
```