```fortran
program secant_method
  implicit none

  real :: x0, x1, x2, tol
  real, external :: f
  integer :: nIter, i

  call getTol(tol)
  call getIter(nIter)
  call getPoints(x0, x1)

  do i = 1, nIter
    call nextX(f, x0, x1, x2)
    call testX2(f, x2, tol, i)
  end do

  if ( abs(f(x2))>tol ) then
    call error(f, x2)
  end if

end program secant_method

subroutine error(f, x2)
  implicit none
  real :: f, x2
  print *, "The program cannot reach to a root value less than tolerance specified"
  print *, "Please, try to increase the number of iteractions"
  print *, "The last value calculated was", x2
  print *, "f(x=last calculated) = ", f(x2)
end subroutine error

subroutine testX2(f, x2, tol, i)
  implicit none
  real :: f, x2, tol
  integer :: i
  if ( abs(f(x2))<=tol ) then
    print *, "The root of f(x) is ", x2
    print *, "f(x=root) = ", f(x2)
    print *, "The number of iteractions used was", i
    call exit()
  end if
end subroutine testX2

subroutine getTol(tol)
  implicit none
  real :: tol
  do while(.true.)
    print *, "Type a tolerance"
    read (*,*) tol
    if ( tol<=0 ) then
      print *, "You should provide a tolerance greather than 0"
    else
      exit
    end if
  end do
end subroutine getTol

subroutine getIter(nIter)
  implicit none
  integer :: nIter
  do while(.true.)
    print *, "Type a number of iteractions"
    read (*,*) nIter
    if ( nIter<=0 ) then
      print *, "You should provide a number of iteractions greather than 0"
    else
      exit
    end if
  end do
end subroutine getIter

subroutine getPoints(x0, x1)
  implicit none
  real :: x0, x1
  print *, 'Type a value for x0'
  read (*,*) x0
  print *, 'Type a value for x1'
  read (*,*) x1
end subroutine getPoints

subroutine nextX(f, x0, x1, x2)
  implicit none
  real :: f, x0, x1, x2
  x2 = x1-f(x1)*(x1-x0)/(f(x1)-f(x0))
  x0 = x1
  x1 = x2
end subroutine nextX

function f(x) result(y)
  implicit none
  real, intent(in) :: x
  real :: y
  y = sin(x*cos(x) - sin(x))
end function f


```