```
c-----------------------------------------------------------
c     Writes a double precision vector to file 'fname'.
c     If fname is the string '-' then the vector is written
c     to standard output.
c-----------------------------------------------------------

      subroutine dvto(fname,v,n)
c-----------------------------------------------------------
c     Arguments:
c
c        fname:  (I)    File name
c        v:      (I)    Vector to be written
c        n:      (I)    Length of vector
c-----------------------------------------------------------
         implicit         none

         integer          getu,        indlnb

         character*(*)    fname
         integer          n
         real*8           v(n)

         integer          ustdout
         parameter      ( ustdout = 6 )

         integer          i,      uto,       rc
```

```
      if( fname .eq. '-' ) then
         uto = ustdout
      else
         uto = getu()
         open(uto,file=fname(1:indlnb(fname)),
   &          form='formatted',iostat=rc)
         if( rc .ne. 0 ) then
            write(0,*) 'dvto: Error opening ',
   &                   fname(1:indlnb(fname))
            return
         end if
      end if

      do i = 1 , n
         write(uto,*) v(i)
      end do

      if( uto .ne. ustdout ) then
         close(uto)
      end if

      return

   end
```

```
c-----------------------------------------------------------
c     Test program for subroutine 'dvto'.
c
c     Program expects two arguments, the name of a file
c     for output ('-' for stdout) and the length of the
c     test vector to be written.
c-----------------------------------------------------------
      program         tdvto

      implicit        none

c-----------------------------------------------------------
c     The integer function 'i4arg' is defined in the
c     'p329f' library.  It takes two arguments, the first
c     is an integer specifying which program argument is
c     to be parsed as an integer, and the second is a
c     default value which will be returned if the argument
c     was not supplied or could not be converted to an
c     integer.
c-----------------------------------------------------------
      integer         iargc,      indlnb,    i4arg

      integer         maxn
      parameter     ( maxn = 100 000 )
      real*8          v(maxn)
      integer         n

      integer         i
      character*256   fname
```

```
c-----------------------------------------------------------
c     Unless exactly two arguments are supplied, print usage
c     message and exit.
c
c     Note the use of the "logical-if" statement (no then)
c-----------------------------------------------------------
      if( iargc() .ne. 2 ) go to 900
      call getarg(1,fname)
      n = i4arg(2,-1)
      if( n .eq. -1 ) go to 900
c-----------------------------------------------------------
c     Limit the value of n
c-----------------------------------------------------------
      n = min(n,maxn)
c-----------------------------------------------------------
c     Define test vector
c-----------------------------------------------------------
      do i = 1 , n
         v(i) = i
      end do
c-----------------------------------------------------------
c     Call the routine ..
c-----------------------------------------------------------
      call dvto(fname,v,n)
c-----------------------------------------------------------
c     Normal exit
c-----------------------------------------------------------
      stop
c-----------------------------------------------------------
c     Usage exit
c-----------------------------------------------------------
 900  continue
         write(0,*) 'usage: tdvto <file name> <n>'
         write(0,*)
         write(0,*) '       Use ''tdvto -'' to write ',
     &              'to standard output'
      stop
      end
```

```
##############################################################
#     Examples of 'tdvto' (tests 'dvto')
##############################################################
gauss 21> tdvto
 usage: tdvto <file name> <n>

        Use 'tdvto -' to write to standard output

gauss 22> tdvto -
 usage: tdvto <file name> <n>

        Use 'tdvto -' to write to standard output

gauss 23> tdvto - 10
    1.000000000000000
    2.000000000000000
    3.000000000000000
    4.000000000000000
    5.000000000000000
    6.000000000000000
    7.000000000000000
    8.000000000000000
    9.000000000000000
    10.00000000000000

gauss 24> tdvto foo 5
gauss 25> more foo
    1.000000000000000
    2.000000000000000
    3.000000000000000
    4.000000000000000
    5.000000000000000
```