```
c-----------------------------------------------------------
c     mysum:  reads numbers one per line from stdin
c     and writes sum on stdout.  Ignores invalid inputs
c     but counts number encountered and reports on stderr.
c-----------------------------------------------------------
      program      mysum

      implicit     none

c-----------------------------------------------------------
c     vi:    Current number read from stdin
c     sum:   Current sum of numbers read
c     rc:    For storing return status from READ
c     nbad:  Count of number of bad inputs
c-----------------------------------------------------------
      real*8       vi,         sum
      integer      rc,         nbad

c-----------------------------------------------------------
c     Initialize ...
c-----------------------------------------------------------
      nbad = 0
      sum  = 0.0d0
```

```fortran
c-----------------------------------------------------------
c     The following construct is roughly equivalent to
c     a while loop, execution keeps returning to the
c     top of the loop until end of file is detected on
c     stdin.
c-----------------------------------------------------------
 100  continue
         read(*,*,iostat=rc,end=200)  vi
         if( rc .eq. 0 ) then
c-----------------------------------------------------------
c           Read a bona fide real*8 value, update sum.
c-----------------------------------------------------------
            sum = sum + vi
         else
c-----------------------------------------------------------
c           Input was invalid.
c-----------------------------------------------------------
            nbad = nbad + 1
         end if
      go to 100
 200  continue
c-----------------------------------------------------------
c     Write sum on standard output.
c-----------------------------------------------------------
      write(*,*) sum
c-----------------------------------------------------------
c     Report # of invalid inputs only if there were some.
c-----------------------------------------------------------
      if( nbad .gt. 0 ) then
c-----------------------------------------------------------
c        Unit 0 is stderr (standard error) on most Unix
c        systems: if you redirect stdin using '>' and this
c        message is tripped, it will still appear on the
c        terminal.
c-----------------------------------------------------------
         write(0,*) nbad, ' invalid inputs'
      end if

      stop
      end
```

```
c-----------------------------------------------------------
c     Less-commented (i.e. more reasonable level of
c     comments) version of mysum.
c-----------------------------------------------------------
c     mysum_s:  reads numbers one per line from stdin
c     and writes sum on stdout.  Ignores invalid inputs
c     but counts number encountered and reports on stderr.
c-----------------------------------------------------------
      program      mysum

      implicit     none

      real*8       vi,         sum
      integer      rc,         nbad

      nbad = 0
      sum  = 0.0d0

 100  continue
         read(*,*,iostat=rc,end=200)  vi
         if( rc .eq. 0 ) then
            sum = sum + vi
         else
            nbad = nbad + 1
         end if
      go to 100
 200  continue

      write(*,*) sum

      if( nbad .gt. 0 ) then
         write(0,*) nbad, ' invalid inputs'
      end if

      stop

      end
```

```
##############################################################
#     Examples of 'mysum' usage
##############################################################
gauss 21> mysum
1
2
8
10
^D
    21.00000000000000

gauss 22> mysum < first100
    5050.000000000000

gauss 23> mysum
12
2
8
a
10
b
^D
    21.00000000000000
            2 invalid inputs

gauss 24> mysum < first100 > mysum_result
gauss 25> more !$
more mysum_result
    5050.000000000000
```