```
PROGRAM stats
!**
!** Purpose:
!**    To calculate mean and the standard deviation of an input
!**    data set, where each input value can be positive, negative,
!**    or zero.
!**

  IMPLICIT NONE

  !*****  Declare the variables used in this program. *****
  INTEGER :: i             !** Loop index
  REAL :: st_dev           !** The standard deviation of the input samples
  REAL :: x                !** An input data value.
  REAL :: x_b,pos_b,neg_b  !** The average of the input samples.

  REAL :: nsum_x = 0,nsum_x2 = 0 !** Summations of the negative values
  REAL :: psum_x = 0,psum_x2 = 0 !** Summations of the positive values
  INTEGER :: n,neg_c=0, &        !**
             pos_c=0             !** The number of data values in the sample
  REAL :: pos_st_dev,neg_st_dev  !** Subset standard deviations

  !** Get the number of points to input.
  PRINT*, 'Enter number of data values in the sample: '
  READ*, n

  !** Check to see if we have enough input data.
  IF ( n < 2 ) THEN !** Insufficient data
    PRINT*, 'At least 2 values must be entered.'
  ELSE !** we will have enough data, so let's get it.

    DO i = 1, n      !** Loop to read input values.
      PRINT*, "Enter data value : "
      READ*, x       !** Read values
      IF (x < 0) THEN     !** Accumulate neg sums
        neg_c=neg_c + 1
        nsum_x  = nsum_x + x
        nsum_x2 = nsum_x2 + x**2
      ELSE IF (x>0) THEN  !** Accumulate pos sums
        pos_c=pos_c + 1
        psum_x  = psum_x + x
        psum_x2 = psum_x2 + x**2
      ENDIF
    END DO

    IF (pos_c >= 2) THEN  !** Enough data so calculate and print
      pos_b = psum_x/pos_c
      pos_st_dev = SQRT((pos_c*psum_x2 - psum_x**2)/pos_c**2)
      PRINT*, "The mean of the positive subset is        :",pos_b
      PRINT*, "The std. deviation for positive subset is :", pos_st_dev
      PRINT*, "The number of positive data values is     :",pos_c
    ELSE
      PRINT*,"Not enough data for positive stats"
    ENDIF

    IF (neg_c >= 2) THEN !** Enough data so calculate and print
      neg_b = nsum_x/neg_c
      neg_st_dev = SQRT((neg_c*nsum_x2 - nsum_x**2)/neg_c**2)
      PRINT*, "The mean of the negative subset is        :",neg_b
      PRINT*, "The std. deviation for negative subset is :", neg_st_dev
      PRINT*, "The number of negative data values is     :",neg_c
    ELSE
      PRINT*,"Not enough data for negative stats"
    ENDIF

    x_b=(nsum_x+psum_x)/n
    st_dev=SQRT((n*(psum_x2+nsum_x2) - (psum_x+nsum_x)**2)/n**2)
```

```
    PRINT*, "The mean of the full data set is          :", x_b
    PRINT*, "The standard dev. of the full data is     :",st_dev
    PRINT*, "The number of data values is               :", n
  END IF

END PROGRAM stats


!   The mean of the positive subset is        :   3.980000
!   The std. deviation for positive subset is :   1.070327
!   The number of positive data values is     :           5
!   The mean of the negative subset is        :  -3.380000
!   The std. deviation for negative subset is :   1.716276
!   The number of negative data values is     :           5
!   The mean of the full data set is          :  0.3000000
!   The standard dev. of the full data is     :   3.948164
!   The number of data values is               :          10
!
!
!
!   The mean of the positive subset is        :   35.00000
!   The std. deviation for positive subset is :   2.203893
!   The number of positive data values is     :           7
!   The mean of the negative subset is        :  -34.00000
!   The std. deviation for negative subset is :   1.673320
!   The number of negative data values is     :           5
!   The mean of the full data set is          :   6.250000
!   The standard dev. of the full data is     :   34.07620
!   The number of data values is               :          12
!
!
!  [1] Arrays are not necessary here as we can create variables to hold
!      the required summations and use the summations to generate the
!      statistics. Once a data item has contributed to the summations
!      there is no need to retain it's value in the code.
!
!  [2] It is always good practice not to use arrays if they are not
!      required. Arrays use up valuable space in memory so if not needed
!      it is prudent to safeguard system resources.

*******************************************
*******************************************

PROGRAM q1partb
  IMPLICIT NONE

  INTEGER :: a=2,b=5,c=4,f=12
  REAL :: d=4,e=6

  PRINT*,(d-b)*a/(c-e)-f/6

END PROGRAM q1partb



! Question One
! ============
!
! [Part a]
!
!   4   5    1    3   6     2
! a + b - (c * d) / f + 5.0**e
!
!   6   3     1     4     2     7   5
! a + f * ( b - c ) / ( d - 3 ) - e * f
```

```
!
!   3     1     6     2     4     5
! b * ( c / d ) - ( f + a ) / 3.0 * e
!
!      1     3   2      4   5   6
! (( f - e ) * 4 ** c ) * a / d - b
!
!   3    2    6   4     1     5
! a ** b ** c + d / ( e + f ) * 2
!
!
! [Part b]
!
! I.     0.5
!
! II.    REAL
!
! III.   Operator    Numeric Result    Data Type
!
!          -          -1.0             REAL
!          -          -2.0             REAL
!          *          -2.0             REAL
!          /           1.0             REAL
!          /           2               INTEGER
!          -          -1.0             REAL

*******************************************
*******************************************
```