```fortran
      PROGRAM TEST
      USE FMZM
      IMPLICIT NONE

!  Least squares fit for the coefficients in the asymptotic series for the Jth harmonic number.

!  H(J) = 1 + 1/2 + 1/3 + ... + 1/J  defines the Jth harmonic number.

!  Find an approximation to H(J) of the form:

!  ln(J) + c(1) + c(2)/J + ... + c(k)/J**(k-1)

!  Integrating 1/x from 1 to J gives ln(J) as a first approximation, and we generate N data
!  points (x(i),y(i)) where x(i) is J and y(i) is H(J) for various J values.  Then we do a
!  least squares fit of the model function c(1) + c(2)/J + ... + c(k)/J**(k-1) to the data
!  (x(i),y(i)-ln(i)).

!  Since this is a sample problem, we can compare the results of the fit to the "true"
!  asymptotic formula, where c(1) = 0.57721566..., Euler's constant, and for i > 1,
!  c(i) = -B(i-1)/(i-1).  The B values are Bernoulli numbers, and the first few are:
!  B(1) = -1/2, B(2) = 1/6, B(4) = -1/30, B(6) = 1/42, ..., with the others being zero:
!  B(3) = B(5) = B(7) = ... = 0.

!  The first c's in the list of fitted coefficients give the most agreement with the
!  theoretical values, and the last ones the least.  The linear system is ill-conditioned,
!  but by using high precision we can get good accuracy for several coefficients.
!  For example, using 400 digit precision, 60 data points at intervals of 100 (i.e.,
!  x(i) = 100, 200, 300, ..., 6000), and fitting 60 coefficients, we get at least 50
!  decimal agreement between the fitted c's and the theoretical ones for c(1), ..., c(29).
!  c(41) agrees to 16 decimals, and because the number is large this is 31 significant
!  digit agreement.

      INTEGER :: J, K, N, NGAP
      TYPE (FM) :: H_N, ONE, DET
      TYPE (FM), ALLOCATABLE :: A(:,:), B(:), C(:), X(:), Y(:)
      TYPE (FM), EXTERNAL :: F

!             This is not a good way to compute Euler's constant, but with 150 digit precision,
!             N = 40 data points at intervals of NGAP = 10, fitting K = 40 coefficients we get
!             c(1) = .57721566490153286060651209008240243104215933593992,
!             correct to 50 places.

!             Set FM precision.

      CALL FM_SET(150)

!             N is the number of harmonic data points.

      N = 40

!             NGAP is the gap between harmonic data points.

      NGAP = 10

!             K is the number of coefficients to fit.

      K = 40
```

```
      ALLOCATE(A(K,K),B(K),C(K),X(N),Y(N),STAT=J)
      IF (J /= 0) THEN
          WRITE (*,"(/' Error in HFIT.  Unable to allocate arrays with K,N = ',2I8/)") K,N
          STOP
      ENDIF

!             Generate the harmonic data points.
!             Since the coefficient of the first term in the model, ln(x), is assumed
!             to be 1 and is not being fitted, subtract that from the Y data points.

      H_N = 0
      ONE = 1
      WRITE (*,*) ' '
      WRITE (*,*) ' Data points:'
      WRITE (*,*) ' '
      DO J = 1, N*NGAP
         H_N = H_N + ONE/J
         IF (MOD(J,NGAP) == 0) THEN
             X(J/NGAP) = J
             Y(J/NGAP) = H_N - LOG(X(J/NGAP))
             WRITE (*,"(A,I4,A,I6,A,A)") ' I = ',J/NGAP,'   X = ',J,'   Y = ',  &
                                          TRIM(FM_FORMAT('F40.35',Y(J/NGAP)))
         ENDIF
      ENDDO

!             Generate the linear system for the normal equations.

      CALL FM_GENEQ(F,A,B,K,X,Y,N)

!             Solve the linear system for the normal equations.

      CALL FM_LIN_SOLVE(A,C,B,N,DET)

!             Print the solution.
!             When using F format, FM doesn't like to print 0.00000...0 showing no
!             significant digits when the actual number is too small for that format.
!             FM will shift to E format when possible, to avoid showing all zeroes.
!             In this example, all the even-numbered coefficients are zero in the
!             asymptotic series for the harmonic numbers, so any non-zero digits
!             found in the fit are not interesting.  Therefore the if statement
!             below prints exactly zero when C(J) is too small, making the output
!             look neater.

      WRITE (*,*) ' '
      WRITE (*,*) ' Fitted coefficients:'
      DO J = 1, K
         IF (ABS(C(J)) > 1.0D-50) THEN
             WRITE (*,"(A,I3,A,A)") ' J = ',J,' C(J) = ',TRIM(FM_FORMAT('F60.50',C(J)))
         ELSE
             WRITE (*,"(A,I3,A,A)") ' J = ',J,' C(J) = ',TRIM(FM_FORMAT('F60.50',TO_FM(0)))
         ENDIF
      ENDDO

      END PROGRAM TEST

      FUNCTION F(J,X)     RESULT (RETURN_VALUE)
      USE FMZM
```

```
      IMPLICIT NONE

!  This defines the model function being fitted to the data points.
!  For the harmonic number case, the model function is:

!  F(J,X) = 1/X**(J-1)

!  This will fit the terms  c1 + c2/n + c3/n**2 + ... to the harmonic model function
!  ln(x) + c1 + c2/n + c3/n**2 + ....

      INTEGER :: J
      TYPE (FM) :: RETURN_VALUE, X

      RETURN_VALUE = 1/X**(J-1)

      END FUNCTION F
```