```
      PROGRAM TEST

!  One use for FM involves programs that don't need multiple precision results but do need some
!  of the special functions available in FM but not in the Fortran standard.  These include:

!  BERNOULLI(N)
!  BETA(X,Y)
!  BINOMIAL(N,K) or BINOMIAL(X,Y)
!  COS_INTEGRAL(X)
!  COSH_INTEGRAL(X)
!  EXP_INTEGRAL_EI(X)
!  EXP_INTEGRAL_EN(N,X)
!  FRESNEL_C(X)
!  FRESNEL_S(X)
!  INCOMPLETE_BETA(X,A,B)
!  INCOMPLETE_GAMMA1(X,Y)
!  INCOMPLETE_GAMMA2(X,Y)
!  LOG_INTEGRAL(X)
!  POCHHAMMER(X,N)
!  POLYGAMMA(N,X)
!  PSI(X)
!  SIN_INTEGRAL(X)
!  SINH_INTEGRAL(X)

!  See the complete list of FM functions in FM_User_Manual.txt.

!  For this application, no TYPE(FM) variables need to be declared.  Just add USE FMZM at the top
!  and compile and link the program like SampleFM.f95.

      USE FMZM
      IMPLICIT NONE

      INTEGER :: J
      DOUBLE PRECISION :: A, B, C, C_FM, ERR, MAX_ERR

!             To use with 53-bit double precision, having about 16 significant digits of accuracy,
!             set the FM precision to 16 digits.

      CALL FM_SET(16)


!             1.  Check to see if Fortran's intrinsic gamma function is correctly rounded.

!                 A is the double precision variable, so GAMMA(A) uses Fortran's intrinsic gamma.

!                 TO_FM(A) converts A to an FM number, so GAMMA( TO_FM(A) ) uses FM's gamma,
!                 then the "=" rounds the result back to double precision variable C_FM.

!                 It is possible that different compilers might give different results for this
!                 test.  Some compilers may not give results that are correctly rounded to full
!                 double precision accuray when A is large, but C_FM should be correctly rounded.

      MAX_ERR = 0
      DO J = 10, 150, 10
         A = J + 0.5D0
         C = GAMMA(A)
         C_FM = GAMMA( TO_FM(A) )
```

```
          ERR = ABS( (C - C_FM) / C_FM )
          IF (ERR > MAX_ERR) THEN
              MAX_ERR = ERR
              B = A
          ENDIF
      ENDDO

      WRITE (*,"(///A/)") " Sample 1.  Compare Fortran's built-in gamma function to FM's"
      IF (MAX_ERR > 0) THEN
          A = B
          WRITE (*,"(A,ES13.7,A,F7.3)") ' Maximum relative error in Fortran gamma was ',  &
                                        MAX_ERR, ' for A = ', A
          C = GAMMA(A)
          WRITE (*,"(ES25.15,A)") C,     ' = GAMMA(A)'
          C_FM = GAMMA( TO_FM(A) )
          WRITE (*,"(ES25.15,A)") C_FM, ' = GAMMA( TO_FM(A) )'
      ELSE
          WRITE (*,"(A)") ' All Fortran gamma results were correctly rounded.'
      ENDIF


!             2.  Binomial coefficients.

!                 Find the probability of getting exactly 10,000 heads in 20,000 tosses
!                 of a fair coin.

!                 Here we could not store the results of the binomial and power separately in
!                 double precision, since BINOMIAL( 20000, 10000 ) = 2.2e+6018 and
!                 2**20000 = 4.0e+6020 would both overflow in double precision.

      WRITE (*,"(//A)") " Sample 2.  Binomial coefficients"
      WRITE (*,"(A)")   "            Find the probability of getting exactly 10,000 heads"
      WRITE (*,"(A/)")  "            in 20,000 tosses of a fair coin."

      C_FM = BINOMIAL( TO_FM(20000), TO_FM(10000) ) / TO_FM(2)**20000

      WRITE (*,"(A,F20.16)") " BINOMIAL( TO_FM(20000), TO_FM(10000) ) / TO_FM(2)**20000 =", C_FM


!             3.  Log Integral function.

!                 Estimate the number of primes less than 10**30.

      WRITE (*,"(//A)") " Sample 3.  Log integral"
      WRITE (*,"(A/)")  "            Estimate the number of primes less than 10**30."

      C_FM = LOG_INTEGRAL( TO_FM('1.0E+30') )

      WRITE (*,"(A,ES23.15)") " LOG_INTEGRAL(TO_FM('1.0E+30')) =", C_FM


!             4.  Psi and polygamma functions.

!                 Rational series can often be summed using these functions.
!                 Sum (n=1 to infinity) 1/(n**2 * (8n+1)**2) =
!                 16*(psi(1) - psi(9/8)) + polygamma(1,1) + polygamma(1,9/8)
!                 Reference: Abramowitz & Stegun, Handbook of Mathematical Functions,
!                 chapter 6, Example 10.
```

```
      WRITE (*,"(//A)") " Sample 4.  Psi and polygamma functions."
      WRITE (*,"(A)")   "            Sum (n=1 to infinity) 1/(n**2 * (8n+1)**2) ="
      WRITE (*,"(A/)")  "            16*(psi(1) - psi(9/8)) + polygamma(1,1) + polygamma(1,9/8)"

      C_FM = 16*( PSI( TO_FM(1) ) - PSI( TO_FM(9)/8 ) ) +           &
             POLYGAMMA( 1, TO_FM(1) ) + POLYGAMMA( 1, TO_FM(9)/8 )

      WRITE (*,"(A,F19.16)") " Sum =", C_FM


!             5.  Incomplete gamma and gamma functions.

!                 Find the probability that an observed chi-square for a correct model should be
!                 less that 2.3 when the number of degrees of freedom is 5.
!                 Reference: Knuth, Volume 2, 3rd ed., Page 56, and Press, Flannery, Teukolsky,
!                 Vetterling, Numerical Recipes, 1st ed., Page 165.

      WRITE (*,"(//A/)") " Sample 5.  Incomplete gamma and gamma functions."

      C_FM = INCOMPLETE_GAMMA1( TO_FM(5)/2, TO_FM('2.3')/2 ) / GAMMA( TO_FM(5)/2 )

      WRITE (*,"(A,F19.16/)") " Probability =", C_FM


      END PROGRAM TEST
```