1. ¿ Qué resultado produce el siguiente programa al ejecutarlo ?. Explicar porqué.

```
      X=0.0
      H=0.1
      DO 10 I=1,10
         X=X+H
 10   CONTINUE
      Y=1.0-X
      WRITE(6,20) X,Y
 20   FORMAT(2E20.10)
      STOP
      END
```

2. Considere los dos siguientes programas FORTRAN

```
      EPS = 1.
 10   EPS=EPS/2.
      WRITE(6,20) EPS
 20   FORMAT(G20.10)
      EPSP1=EPS+1
      IF(EPSP1.GT.1.) GOTO 10
      STOP
      END
```

```
      EPS = 1.
 10   EPS=EPS/2.
      WRITE(6,20) EPS
 20   FORMAT(G20.10)
      IF(EPS.GT.0.) GOTO 10
      STOP
      END
```

Ejecute los programas y explique los resultados.

3. ¿Qué resultado produciría el siguiente programa FORTRAN cuando se ejecuta en diferentes ordenadores?. Intente predecir la salida antes de ejecutar el programa; entonces ejecútelo y confirme las respuestas.

```
      EPS=1.
      KBIT=0
 10   EPS=.5*EPS
      KBIT=KBIT+1
      EPSP1=EPS+1.
      IF((EPSP1.GT.1.).AND.(EPSP1-EPS.EQ.1.))GOTO 10
      IF (EPSP1-EPS.EQ.1.) EPS=2.*EPS
      WRITE(6,20)EPS,KBIT
      STOP
 20   FORMAT(G20.10,I10)
      END
```

¿Cuál es el efecto de incluir `DOUBLE PRECISION EPS,EPSP1` al comienzo del programa?

4. Los compiladores FORTRAN no proporcionan en general una función para evaluar la cotangente. Podemos escribir una propia usando `COT(X)=COS(X)/SIN(X)`. Existen varias formas de chequear la exactitud; un test simple es comparar `COT(X)` con `(COT(X/2)-1.0/COT(X/2))/2.0`, que es matemáticamente igual a la cotangente. Escribir un programa para evaluar ambas funciones para 2000 puntos en el intervalo $(6\pi, 25\pi/4)$. Imprimir el error medio y el error máximo. ¿Son satisfactorios los resultados? ¿Los errores que se observan son del cálculo de `COT(X)` o de la resta de la segunda expresión? Para chequear esto último use doble precisión.

5. Considere la función FORTRAN :

```
      REAL FUNCTION F(X,N)
      REAL X
      F=ABS(X)
      DO 10 I=1,N
         F=SQRT(F)
 10   CONTINUE
      DO 20 I=1,N
         F=F*F
 20   CONTINUE
      RETURN
      END
```

Si no existiesen errores de redondeo obtendríamos `F=ABS(X)` pero esto no es lo que ocurre. Experimente con ésta función para varios valores de x, tanto $|x| < 1$ como $|x| > 1$. Explique los resultados.

6. Considere un sistema numérico F en coma flotante con una mantisa de 3 dígitos decimales, usando cortado (chopping) en lugar de redondeo (rounding):

$$x = \pm .d_1 d_2 d_3 \times 10^e \qquad \text{con} \qquad -100 \le e \le 100 \qquad \text{y} \qquad 0 \le d_i \le 9.$$

Supongamos que F sea un sistema normalizado (o sea, $d_1 \neq 0$ salvo que $x = 0$). El número cero está contenido en F y tiene la representación única como $+.000 \times 10^{-100}$. Denotemos con $\oplus$ la operación suma en coma flotante : $\forall x, y \in F$, el valor de $x \oplus y$ se define como el número en coma flotante más próximo a $x+y$ cuya magnitud es menor o igual a la magnitud de $x+y$. Denotemos con $\otimes$ la operación multiplicación en coma flotante : $\forall x, y \in F$, el valor de $x \otimes y$ se define como el número en coma flotante más próximo a $x \times y$ cuya magnitud es menor o igual a la magnitud de $x \times y$.

- ¿Cuántos números reales diferentes pueden ser representados exactamente por F?
- Encontrar ejemplos de x, y, z en Fque prueben que las siguientes afirmaciones *no son* generalmente ciertas, incluso si el resultado está dentro del rango de F :

$$(1)(x \oplus y) \oplus z = x \oplus (y \oplus z) \qquad (2)(x \otimes y) \otimes z = x \otimes (y \otimes z)$$

7. El algoritmo $x_{n+1} = (x_n + a/x_n)/2$ produce la raíz cuadrada de a. Aplicarlo con una aproximación inicial $x_0 = a/2$. Comparar los resultados con los de la función intrínseca `SQRT(X)`.

8. Escribir una subrutina para formar la suma $S = \sum_1^n a_i$ de tres formas : (1) del elemento más pequeño al mayor, (2) del mayor elemento al más pequeño y (3) en doble precisión con conversión a simple precisión al final de la suma. Use el resultado en doble precisión para encontrar el error en los dos resultados en precisión simple. Imprima los resultados. Escribir también un programa principal para crear las siguientes series en precisión simple y que use la subrutina para sumar las series.

$$(a)\sum_1^n \frac{1}{j} \qquad (b)\sum_1^n \frac{1}{j^2} \qquad (c)\sum_1^n \frac{1}{j^3} \qquad (d)\sum_1^n \frac{(-1)^j}{j} \qquad (e)1 + \sum_1^n \frac{1}{j^2 + j}$$

9. Obtener una fórmula recursiva para calcular las integrales :

$$y_n = \int_0^1 \frac{x^n}{4x+1} dx$$

dando un algoritmo que va bien y otro que funciona de forma pobre, ambos basados en la fórmula de recursión.

10. Contar el número de multiplicaciones, sumas y divisiones requeridas por la regla de Ruffini-Horner y por el siguiente algoritmo alternativo de Shaw-Traub (1974) :

Algoritmo 1 ALGORITMO DE SHAW-TRAUB

calcular y almacenar $x_0^2, \ldots, x_0^n$
para $0 \leq j \leq n$ **hacer**
  $b_j \leftarrow a_j x_0^j$
fin para
para $k = 0, 1, \ldots, n-1$ **hacer**
  para $j = n-1, \ldots, k+1, k$ **hacer**
    $b_j \leftarrow b_j + b_{j+1}$
  fin para
fin para
para $0 < j \leq n$ **hacer**
  $b_j \leftarrow b_j / x_0^j$
fin para

11. Queremos calcular la expresión $(\sqrt{2}-1)^6$ usando el valor aproximado 1.4 para $\sqrt{2}$. Podemos elegir entre sustituir éste valor aproximado en la expresión anterior o en cualquiera de las siguientes :

$$\frac{1}{(\sqrt{2}+1)^6}, \quad (99-70\sqrt{2}), \quad \frac{1}{99+70\sqrt{2}}, \quad (3-2\sqrt{2})^3, \quad \frac{1}{(3+2\sqrt{2})^3}$$

¿ Cuál de las seis alternativas da el mejor resultado?

12. Escribir una función lógica F77 que devuelva si un año es bisiesto : si el año es divisible por 4 es bisiesto, pero si también es divisible por 100 no es bisiesto, salvo si es divisible por 400.

13. Realizar programas/subrutinas F77 para
    (a) Sumar dos polinomios
    (b) restar dos polinomios
    (c) multiplicar dos polinomios
    (d) dividir dos polinomios
    (e) potencia entera de un polinomio (*)

14. Realizar programas/subrutinas F77 para
    (a) Sumar dos vectores
    (b) restar dos vectores
    (c) hallar el producto de un escalar por un vector
    (d) producto escalar de dos vectores
    (e) hallar la norma de un vector

15. Realizar programas/subrutinas F77 para
    (a) Sumar dos matrices
    (b) restar dos matrices
    (c) hallar el producto de un escalar por una matriz

    (d) multiplicar dos matrices
    (e) potencia entera de una matriz (*)
    (f) hallar la norma de una matriz

16. Obtener las raíces de un polinomio de grado dos $ax^2 + bx + c$.

17. Obtener las raíces de un polinomio de grado tres $ax^3 + bx^2 + cx + d$.

18. Obtener las raíces de un polinomio de grado cuatro $ax^4 + bx^3 + cx^2 + dx + e$.

19. Escribir programas F77 que calculen las siguientes fórmulas de combinatoria :
    (a) número de combinaciones sin repetición (y con repetición) de n elementos tomados de m en m.
    (b) número de variaciones sin repetición (y con repetición) de n elementos tomados de m en m.
    (c) número de permutaciones sin repetición (y con repetición) de n elementos tomados m_1 iguales, $\cdots$, m_k iguales entre sí.

20. Escribir un programa que demuestre numéricamente que la identidad $\operatorname{sen}^2 x + \cos^2 x = 1$ se cumple para variables complejas.

21. La siguiente serie es válida para $0 < x \leq 2$:

$$\ln x = \frac{x-1}{1} - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \cdots$$

que converge lentamente para valores próximos a cero y a 2, pero para $0.5 \leq x \leq 1.5$ da una exactitud de seis decimales si se usan los términos hasta $-(x-1)^{16}/16$.

Escribir el programa F77 correspondiente y comparar los resultados con los que da la función `LOG`, para argumentos desde 0.5 hasta 1.5 en pasos de 0.1.

22. Programa F77 par cambios de base numérica (bases 2 a 16).

23. El siguiente algoritmo calcula a^n. Obtener su orden $O(n^k)$ y aplicarlo al cálculo de la potencia entera de polinomios y matrices.

Algoritmo 2 ALGORITMO DE POTENCIACIN

$a1 = a, \quad n1 = n$
$x = 1$
mientras $(n1 \neq 0)$ **hacer**
  mientras $(n1 \mod 2 = 0)$ **hacer**
    $n1 = n1$ **div** 2
    $a1 = a1 * a1$
  fin mientras
  $n1 = n1 - 1$
  $x = x * a1$
fin mientras
devolver x