Computación, Práctica 7.

Programación en FORTRAN: Subrutinas.

Ejercicio 1. a) Para una rutina que comienza

```
SUBROUTINE CONF(X, Y, N)
INTEGER X
REAL Y, N
```

indicar la correspondencia entre los argumentos para el llamado

```
CALL CONF(N, Y, X)
```

b) Dadas las siguientes declaraciones,

```
SUBROUTINE CHOP(X, Y, RESULT)
```

y

```
SUBROUTINE FIX(W, 4.0)
```

¿cuál es incorrecta, y por qué?

c) ¿Qué valor de x se imprime?

```
PROGRAM PRUEBA
REAL X
CALL SUB(X)
WRITE(*,*) X
END

SUBROUTINE SUB(Y)
REAL Y, X
X = 25.0
Y = 2.0 * x
RETURN
END
```

Ejercicio 2. Una estrella pertenece a la llamada Secuencia Principal si su luminosidad se incrementa con la masa de acuerdo a la ley

$$\frac{L}{L_\odot} = \left(\frac{M}{M_\odot}\right)^{\frac{7}{2}}, \tag{1}$$

donde $L_\odot$ y $M_\odot$ son la luminosidad y masa del Sol respectivamente. Construir una subrutina que decida si las estrellas, cuyas medidas están en el archivo P07-Tabla.dat, pertenecen o no a la Secuencia Principal y guarde el resultado en un archivo.[1] Consideraremos que una estrella está en la secuencia si el valor de la luminosidad predicho en (1) cae dentro del rango medido. **Nota:** a) Los datos están normalizados, $L_\odot = M_\odot = 1$. b) La rutina puede tener, o no, parámetros de entrada.

Ejercicio 3. a) Escribir una subrutina para multiplicar una matriz $A^{n\times m}$ por un vector B^m, ambos con elementos reales. Verificar, por ejemplo, con

$$AB = T, \qquad A = \begin{pmatrix} 1 & -9 & -4 & 3 \\ -1 & 0 & 5 & 1 \\ 3 & 6 & -4 & 8 \end{pmatrix},$$

$$B = \begin{pmatrix} 2 \\ -1 \\ 0 \\ 5 \end{pmatrix}, \qquad T = \begin{pmatrix} 26 \\ 3 \\ 40 \end{pmatrix}$$

b) Escribir una subrutina para multiplicar dos matrices $C^{n\times m}$ y $D^{m\times l}$, ambas con elementos reales. Verificar, por ejemplo, con

$$CD = U, \qquad C = \begin{pmatrix} 1 & -9 & -4 \\ -1 & 0 & 5 \end{pmatrix},$$

$$D = \begin{pmatrix} 2 & -1 \\ 0 & 5 \\ 3 & -6 \end{pmatrix} \qquad U = \begin{pmatrix} -10 & -22 \\ 13 & -29 \end{pmatrix}$$

c) Escribir una rutina que calcule la potencia p-ésima de una matriz *cuadrada* $E^{n\times n}$, con $p \geq 0 \in \mathbb{Z}$. Para ello definimos $E^0 = I$, $E^p = E * E^{p-1}$. Verificar, por ejemplo, con una matriz de Hankel de orden 5 elevada al cubo

$$E = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 0 \\ 3 & 4 & 5 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E^3 = \begin{pmatrix} 294 & 404 & 455 & 420 & 275 \\ 404 & 509 & 526 & 430 & 200 \\ 455 & 526 & 480 & 294 & 130 \\ 420 & 430 & 294 & 171 & 70 \\ 275 & 200 & 130 & 70 & 25 \end{pmatrix}$$

Ejercicio 4. Realice una subrutina que implemente

[1] Un dato experimental suele escribirse $L \pm \Delta L$. Esto significa que el dato "verdadero" es cualquiera que caiga dentro del rango $[L - \Delta L, L + \Delta L]$. La cuarta columna de la tabla corresponde a los ΔL. Se supone además que las medidas de la masa no tienen error.

el método de la Regula Falsi, detallado en el ejercicio 9 de la Práctica 6. Realice un programa que haga uso de dicha subrutina para calcular los ceros en el intervalo $(0,1)$ de las siguientes funciones:

$f(x) = e^{-x} - 0.5$ 1 raíz, con un error de 10^{-3},

$f(x) = 0.5\,(5\,x^3 - 3\,x)$ 1 raíz, con un error de 10^{-4},

$f(x) = \text{sen}\,(3\,x)\,\ln(x) + 0.5$ 2 raíces, con un error de 10^{-2}.

Imprimir en cada caso el valor de la raíz, su error correspondiente y la cantidad de iteraciones utilizadas. En cada uno de los casos elegir convenientemente el valor inicial usando `gnuplot`. Pasar las funciones a la subrutina como argumentos.

Ejercicio 5. Reescribir el programa del ejercicio 10 de la práctica anterior usando subrutinas y en doble precisión. La función a integrar debe ser pasada a la subrutina como argumento. Estimar el error de la integral calculada con n intervalos usando ahora la expresión $E_n = \frac{S_n - S_{n/2}}{15}$, donde S_n es precisamente la integral que acabamos de calcular y $S_{n/2}$ la integral calculada con $n/2$ intervalos.

Ejercicio 6. Para las matrices, $A^{m\times n}$, y sus respectivas variables de control, v, que se encuentran en el archivo P07-Matrices.dat, escribir un programa que:

1. si v = 'c', intercambie dos a dos (primera con segunda, tercera con cuarta, etc.) las columnas de la matriz.
2. si v = 't', trasponga la matriz.
3. si v = 'd', calcule la traza de la matriz.
4. si v= 'l', genere una matriz lógica con valores `.true.` si $a_{ij} \geq 0$ y `.false.` si $a_{ij} < 0$.

El programa debe imprimir en un archivo de salida la matriz original y el resultado correspondiente. Usar una subrutina para cada opción. ¿En qué casos podrían usarse funciones en vez de subrutinas?

Nota: Definir la menor cantidad posible de arreglos.

Ejercicio 7. La posición de una partícula en función del tiempo está dada por la expresión

$$x(t) = e^{-\frac{1}{2}\,(t-6)^2}\,\left(-1151 + 565\,t - 92\,t^2 + 5\,t^3\right)$$

Escribir un programa que:

a) Determine, e imprima en pantalla, cuánto tiempo demora la partícula en llegar a $x = 2$, midiendo el tiempo a intervalos de 0.1 seg a partir de $t = 0$.

b) Determine, e imprima en pantalla a través de una variable lógica, si la partícula alcanza una velocidad de 12 unidades en el intervalo $[0, 5]$.

c) Guarde en un archivo a 3 columnas, con formato exponencial de 3 decimales, t, $x(t)$ y la velocidad $v(t)$ de la partícula. La escritura debe finalizar cuando, para tiempos mayores a 10 s, $|v(t)| < 10^{-4}$.

Ejercicio 8. En el archivo *P07-Puntos.dat* se encuentran listados una serie de puntos del plano XY generados al azar. Además, sean dos circunferencias, una de centro (-2.0; 0.0) de radio 4 y otra de radio 5 centrada en (2.0; -2.0). Estas circunferencias definen 4 regiones en las cuales se pueden ubicar los puntos listados en el archivo, a saber:

- La región 1 corresponde a los puntos que están dentro de la circunferencia más chica, pero fuera de la más grande.
- La región 2 corresponde a los puntos que están dentro de la circunferencia más grande, pero fuera de la más chica.
- La región 3 corresponde a los puntos que están dentro de las dos circunferencias.
- La región 4 corresponde a los puntos que están fuera de las dos circunferencias.

Realizar un programa que determine el porcentaje de puntos que hay en cada región.

Por otro lado, en un archivo cuyo nombre es dado por el usuario, listar los puntos indicando a que región pertenecen.

Usar una función de sentencia para calcular los porcentajes.

Para determinar a que región pertenece cada punto, construir una función externa que, para un punto dado devuelva el número de la región correspondiente.

Nota: Considerar que un punto sobre una circunferencia está dentro de la misma.

Ejercicio 9. En el archivo *P07-Acciones.dat* se da la cotización de las acciones en el mercado de valores argentino para el 26 de mayo de 2022. La primera y segunda columnas se corresponden con el símbolo y el nombre de la acción, la tercera es el último valor operado, la cuarta la variación diaria, la quinta y la octava corresponden a los valores de apertura y cierre respectivamente, la sexta y séptima a los valores mínimos y máximos alcanzados durante el día y la última al mónto operado del día.

Construir un programa que lea el archivo *P07-Acciones.dat* y lo ordene de mayor a menor según la variación del valor de cada acción. Para ordenar debe usarse una subrutina y el resultado debe escribirse desde el programa principal con formato adecuado en un archivo cuyo nombre estará dado por el usuario.

Por pantalla debe escribirse las acciones de mayor valor en el momento de la apertura, del cierre y de la última cotización. Para ello construir una función externa que para un dado vector devuelva el índice del elemento mayor.

Nota: A priori no se sabe cuántas acciones hay en *P07-Acciones.dat*.