Computación, Práctica 5.

Programación en FORTRAN: DO WHILE y arreglos.

Ejercicio 1. Escribir el siguiente bucle utilizando `DO WHILE`

```
      SUM=0.
      WRITE(*,*) "Ingrese el primer numero"
      READ(*,*) P
9     IF (P.NE.PL) THEN
         SUM=SUM+P
         WRITE(*,*)  "Proximo numero"
         READ(*,*) P
         GOTO 9
      ENDIF
```

Ejercicio 2. Si L es un arreglo de seis elementos reales, ¿qué hacen los siguientes fragmentos de un programa?

a)

```
      DO J = 1, 5
         L(J+1) =L (1)
      ENDDO
```

b)

```
      DO J = 2, 6
         L(J-1) = L(J)
      ENDDO
```

Ejercicio 3. Sean tres arreglos A(5,-5:5,10), B(5,6) y C(0:7,-3:6). ¿Cuántos elementos tiene cada uno? Listar, tal como están guardados en memoria, los seis primeros elementos y el último elemento del arreglo B.

Ejercicio 4. El vector B está compuesto de M números complejos. Escribir un programa que encuentre el de mayor módulo. Imprimir el índice, el número complejo y el valor del módulo encontrado. Probarlo con un vector de 5 elementos.

Ejercicio 5. Sea $Q(N)$ un vector de números reales dado por el usuario. Calcular el promedio $\bar{X} = \frac{1}{N}\sum_{j=1}^{N} Q_j$ y la desviación standard $\sigma = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(Q_j - \bar{X})^2}$. Imprimir los datos pedidos y una tabla de tres columnas: el índice, las componentes del vector, y las diferencias entre las mismas y el valor medio calculado.

Ejercicio 6. Usar un arreglo de caracteres para guardar un conjunto de doce palabras, donde cada palabra tiene un máximo de 10 caracteres. Una vez ingresadas las palabras, el programa debe imprimir cada una de ellas, el número de letras que poseen y el número de palabras de cuatro letras que fueron ingresadas. Para determinar el número de letras de cada palabra utilice la función `INDEX`.

El "método de la burbuja" es un algoritmo para ordenar listas de números. Consiste en comparar elementos adyacentes de la lista e intercambiarlos si están desordenados. Esquemáticamente, el algoritmo es:

```
while el vector no esté ordenado do
  do para cada par de elementos adyacentes
    if los valores del par están desordenados then
      intercambiar los valores
    end if
  end do
end while
```

Ejercicio 7. Realice un programa que ordene las componentes de los vectores dados en el archivo *P05-Vectores.dat* en forma decreciente utilizando el método de la burbuja. El vector ordenado debe ser almacenado en el arreglo original, es decir, no usar vectores adicionales.

Ejercicio 8. Realizar un programa que multiplique dos matrices A y B, las cuales, al igual que sus dimensiones puedan ser entradas por teclado. Verificar con un ejemplo sencillo.

Ejercicio 9. La solución al sistema de ecuaciones lineales

$$\begin{aligned} a_{11}\,x_1 &= b_1 \\ a_{21}\,x_1 + a_{22}\,x_2 &= b_2 \\ \cdots &= \cdots \\ a_{n1}\,x_1 + a_{n2}\,x_2 + a_{n3}\,x_3 + \cdots + a_{nn}\,x_n &= b_n \end{aligned}$$

donde la matriz de coeficientes $A \in \mathbb{R}^{n\times n}$ es triangular inferior (los coeficientes que no aparecen en el sistema son nulos), puede ser encontrada mediante la fórmula:

$$x_j = \frac{1}{a_{jj}}\left(b_j - \sum_{k=1}^{j-1} a_{jk}\,x_k\right) \qquad j = 1, \ldots, n.$$

Escribir un programa para este método, y verificar su correcta implementación con el siguiente ejemplo:

$$\begin{pmatrix} -5 & & & \\ -1 & -2 & & \\ -1 & 3 & 2 & \\ 4 & 2 & 3 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -15 \\ -7 \\ 5 \\ 10 \end{pmatrix}$$

cuya solución es: $x_1 = 3.0$, $x_2 = 2.0$, $x_3 = 1.0$, $x_4 = 9.0$.

El programa debe dar las dimensiones de matrices y vectores necesarios via `PARAMETER` y debe terminar con aviso de error en caso que algún elemento de la diagonal sea nulo.

Ejercicio 10. Escribir un programa que, dada una matriz $A^{m\times n}$, invierta el orden de los coeficientes en cada una de las columnas (cambiar primer coeficiente con el último, segundo con el anteúltimo, etc) sin usar ni vectores ni matrices auxiliares.

Fijar las dimensiones de la matriz via `PARAMETER`, y definirla usando la sentencia `DATA`. Se debe imprimir en pantalla la matriz antes y después del cambio.

Ejercicio 11. Sea $Q \in \mathbb{C}^{m\times n}$ una matriz compleja. Escribir un programa tal que si la suma de las partes reales de los elementos de la diagonal es negativa (comparar m y n para hacer la suma), construya una matriz de caracteres, con coeficientes iguales a 'N' si la parte imaginaria del coeficiente de Q correspondiente es negativa, y a 'P' si es positiva o nula; en caso de ser la suma de las partes reales de los elementos de la diagonal positiva o nula, debe ser construída una matriz lógica, con coeficientes de valor *.FALSE.* cuando el correspondiente de Q tenga módulo menor a uno, y *.TRUE.* en caso contrario.

El programa debe imprimir en pantalla el valor de la suma de las partes reales de la diagonal, y la matriz obtenida. Usar la matriz del archivo *P05-Matriz.dat* para correr un ejemplo.

Anteriormente hemos visto, en la práctica 3, cómo utilizar la sentencia `END` cuando leíamos un archivo del que desconocíamos el número total de filas a leer. Una forma más moderna utilizando `DO WHILE` es la que mostramos en el siguiente ejemplo:

```
N = 0
IO = 0
DO WHILE (IO .GE. 0)
   READ(10,*,IOSTAT=IO) A, B, C
   IF (IO .EQ. 0) THEN
      N = N + 1
   ENDIF
ENDDO
```

¿Qué significado tiene un valor no nulo en la variable `IOSTAT`?

Ejercicio 12. Escribir un programa que lea el archivo *P03-Puntos.dat* del ejercicio 8 de la práctica 3, y determine cuántos datos contiene, usando el algoritmo anterior.

Programando en Fortran 90, ejercicios adicionales:

Ejercicio 13. En el archivo *P05-Matrices.dat* hay dos matrices con sus correspondientes dimensiones. Realizar un programa que lea la primera matriz utilizando asignación dinámica de memoria y la multiplique por la matriz B dada más abajo utilizando la función `MATMUL`. La matriz B debe ser asignada en el programa mediante la función `RESHAPE`. Escribir con un formato adecuado la matriz resultante.

Liberar la memoria y en el mismo arreglo que usamos para la primera matriz leer la segunda matriz.

Usando la función `DOT_PRODUCT` realizar el producto escalar de la segunda fila por la tercera columna.

$$B = \begin{pmatrix} -5 & 0 & 1 & 2 \\ -1 & -2 & 0 & 0 \\ -1 & 3 & 2 & 1 \\ 4 & 2 & 3 & -1 \end{pmatrix}$$