

5.1.- LAZOS

Hasta el momento los contenidos desarrollados en los capítulos anteriores nos han permitido conocer dos de los tres tipos de estructuras fundamentales en cualquier tipo de lenguaje de programación que son:

ESTRUCTURA 1 (SECUENCIA): Grupo de sentencias que se van ejecutando de manera secuencial.

ESTRUCTURA 2 (SELECCION): Bloque de bifurcación que permite ejecutar diferentes grupos de sentencias.

Ahora se describirá el tercer tipo fundamental de estructura de cualquier programa, el LAZO o BUCLE.

ESTRUCTURA 3 (LAZO o BUCLE): Permite la repetición de cualquiera de las dos estructuras anteriores de manera cíclica.

Este tipo de construcciones nos permite realizar una operación o grupo de sentencias un número de veces que se fijara a través de una variable entera. Su construcción básica se muestra a continuación:

ESTRUCTURA	SIGNIFICADO
<p>DO i = u₁, u₂, u₃ <i>Grupo de sentencias</i></p> <p>ENDDO</p>	<p>DO : Sentencia FORTRAN de comienzo de lazo. i : Índice entero del lazo DO, también se llama <i>variable del lazo DO</i> u₁: Numero entero o real que indica el valor inicial para el LAZO u₂: Numero entero o real que indica el valor final para el LAZO u₃: Numero entero o real que indica el paso o incremento para el LAZO, si se omite el incremento será por defecto u₃ = 1 ENDDO : Sentencia de fin de lazo.</p>

La estructura de LAZO o BUCLE anterior se ejecuta del siguiente modo:

Cuando el programa llega a una sentencia DO comienza una ejecución cíclica en la que la variable **i** irá tomando valores que dependerán de las variables **u₁, u₂, u₃**. Veamos esto para cada ciclo de ejecución:

1º Ciclo: Nada más iniciarse este ciclo la variable **i** tomará el valor **u₁**

2º Ciclo: Al iniciarse el segundo ciclo la variable **i** tomará el valor **u₁ + u₃**

3º Ciclo: Al iniciarse el tercer ciclo la variable **i** tomará el valor **u₁ + 2u₃**

.....

Nº Ciclo: Este será el último ciclo, el cual se produce cuando **i** toma el valor **u₂**.

En caso de que la suma de incrementos no concuerde con el valor de **u₂**, el lazo finalizará con el valor de **i** previo al que sobrepase el valor de **u₂**. Veamos algunos ejemplos sencillos de aplicación:

Ejemplo

LAZO	SALIDA POR PANTALLA
<p>DO i=4,10,2 PRINT*,i ENDDO</p>	<p>4 6 8 10</p>
<p>DO i=30,15,-5 PRINT*,i ENDDO</p>	<p>30 25 20 15</p>

Y deseamos obtener una serie de puntos para realizar una gráfica para un intervalo [A,B]. Se pide realizar un programa que pregunte al usuario los valores del intervalo A y B y un incremento para que el programa pueda establecer un numero finito de valores de la variable X y así obtener los valores de F(x) correspondientes. Además el programa deberá presentar los resultados en dos columnas como se muestra a continuación:

X	f(x)
Valor 1	Resultado 1
Valor 2	Resultado 2
Valor 3	Resultado 3
Valor 4	Resultado 4
.....	
.....	

EJERCICIO 5.4

- Realizar un programa que presente como resultados la equivalencia entre grados y radianes de 0 a 2pi de grado en grado.

5.2.- ANIDAMIENTO DE LAZOS

Al igual que ocurría con la sentencia **IF-ELSE-ENDIF** para la cual estaba permitido ir anidando los bloques de manera continua , siempre y cuando mantuviéramos las reglas de escritura correspondientes, para el caso del bloque **DO-ENDDO** ocurre lo mismo, a continuación se muestra este tipo de estructura.

TIPICO ANIDAMIENTO
<pre style="margin: 0;"> DO i=1,3,1 Sentencias FORTRAN 1º nivel DO j=1,2,1 Sentencias FORTRAN 2º nivel DO k=1,2,1 Sentencias FORTRAN 3º nivel ENDDO ENDDO ENDDO </pre>

Respecto al anidamiento anterior es preciso observar los siguientes aspectos:

- Es muy conveniente realizar un sangrado regular para cada uno de los bloques **DO-ENDDO**, esto facilita la comprensión de la estructura.
- Al estar anidados tres lazos DO aparecen tres variables de lazo **i,j** y **k**. Todas ellas deberán estar declaradas como enteras en la sección de declaración de variables.
- Para entender el comportamiento de este tipo de estructuras anidadas es preciso entender que:

Las sentencias FORTRAN 1º nivel serán ejecutadas 3 veces
Las sentencias FORTRAN 2º nivel serán ejecutadas 3x2=6 veces
Las sentencias FORTRAN 3º nivel serán ejecutadas 3x2x2=12 veces

- El orden en que los ciclos se van produciendo puede entenderse observando los valores que van tomando de **i,j** y **k** a medida que la estructura se ejecuta.

i	j	k
1	1	1
1	1	2
1	2	1
1	2	2
2	1	1
2	1	2
2	2	1
2	2	2
3	1	1
3	1	2
3	2	1
3	2	2

Como puede observarse los lazos mas profundos dentro del anidamiento son los que mas repeticiones realizan en función de los lazos menos profundos. Es decir, para que el índice **j** de un paso ha de completarse el numero total de ciclos establecidos para **K** y de forma equivalente para que **i** de un paso habrán de completarse los ciclos establecidos para **j**

EJERCICIO 5.5

- Realizar un programa con cuatro lazos DO-ENDDO anidados y que liste por pantalla cada una de las variables de lazo establecidas, los resultados deberán ser mostrados de forma equivalente a los a los que muestra la tabla anterior.

EJERCICIO 5.6

- Calcular el valor de la serie de Taylor para e^x y para un numero de términos N introducido por teclado.

$$e^x = 1 + x + (x^2 / 2!) + (x^3 / 3!) + \dots$$

EJERCICIO 5.7 (opcional)

- Calcular el valor de la serie de Taylor para $\sin x$ y para un numero de términos N introducido por teclado.

$$\sin x = x - (x^3 / 3!) + (x^5 / 5!) - (x^7 / 7!) + \dots$$

5.3.- SENTENCIA DO WHILE

Un caso especial de utilización de la sentencia DO es el bloque constituido por las sentencia **DO WHILE-ENDDO**. Este tipo de sentencia tiene por objeto establecer un lazo de ejecución que una vez iniciado solamente finalizara en caso de cambiar de valor la expresión lógica que lo controla. La estructura básica de un lazo DO WHILE-ENDDO se muestra a continuación:

DO WHILE (expresión lógica)

*Grupo sentencias ejecutadas de manera cíclica hasta que la expresión lógica tome el valor **.false**.*

