**CYCLE: se salta el END DO más próximo cuando una expresión se cumple y empieza la siguiente expresión.**

**DO i=1,n**

**DO**

**IF(exp. Log) CYCLE**

**…**

**End do se salta este DO y empieza el siguiente (si es que se cumple)**

**……**

**……**

**END DO**

**Estructuras de REpetición (loops)**

Repetición de sección del código

Es una manera muy común de aproximar soluciones mediante la repetición de un algoritmo

**LOOPS “Mientras”**

Este tipo de loops se utiliza cuando se requiere repetir un bloque de instrucciones pero se desconoce la cantidad de veces que debe repetirse.

**No se sabe la cantidad de veces que se repetirá el código**

I = 1

suma = 1

DO

IF (i>5) EXIT El bloque de instrucciones se repite mientras la expresión lógica sea falsa

suma = suma + 1 En el momento que la expresión se vuelve verdadera se sale del LOOP

i = i +1

END DO

En versiones más modernas de Fortran (95+) se introdujo un LOOP “mientras” equivalente al anterior

i = 1

suma = 1

DO WHILE (i <= 5) El bloque de instrucciones se repite mientras la exp. lógica es verdadera

suma = suma + i En el momento que la expresión se vuelva falsa, el LOOP se detiene

i = i +1

END DO

**LOOPS iterativos**

Cuando se quiere repetir un bloque de instrucciones y se sabe desde el principio cuántas veces debemos repetirlo, conviene usar un loop iterativo.

Ciclos controlados desde el inicio

**Se conoce de entrada la cantidad de veces que se repetirá el código.**

Estructura:

suma = 1

DO i = inicio(1), final(5), incremento(valor índice)(1)

Basándose en los ejemplos anteriores.

Cuando el incremento es de 1 en 1 generalmente no se pone

suma = suma + i

END DO

Cada vez que se va a realizar una repetición se realiza la siguiente expresión:

**i\*incremento <= final\*incremento**

En el momento en que el resultado sea falso se termina el LOOP

Mientras sea verdadero seguirán los LOOP

EJEMPLO:

suma = 1

DO i = 1, 3

suma = suma + i

END DO

Verificar si arranca o no

1\*1 ? 3\*!

1 < 3

suma = 2

2\*1 ? 3\*1

2 < 3

suma = 4

3\*1 ? 3\*1

3 < 3

suma = 7

**4\*1 ? 3\*1**

**4 < 3**

**Se sale del LOOP**

**suma = 7**

¿Cuantas repeticiones se realizan en un LOOP de este estilo?

**Nrepeticiones = INT[(final-inicio+incremento)/incremento]**

En el caso de DO i= 1, 3, 1

...

END DO

Nrepeticiones = (3-1+1)/1 = 3

DO i= 1,7,2

…

END DO

Nrepeticiones= (7-1+2)/2 = 4

Incrementos negativos

suma = 1

DO i=5,0,-1

suma = suma + 1

END DO

5\*(-1) 0\*(-1)

-5 < 0

Se cumple la condición necesaria

Nrepeticiones = (0-5-1)/-1 = -6/-1 = 6

**¡¡Los índices deben de ser de tipo INTEGER!!**

LOOPS “mientras”

i = 1

DO

…

i = i + 1

END DO

En los LOOPS iterativos está prohibida la redefinición del índice dentro del mismo LOOP

**EJEMPLO**

DO i=1,3,1

…

i=4

END DO

**ALGORITMO DE HORACIO**

**Primero hay que hacer un programa principal en donde se le pide al usuario el valor de n**

**PROGRAM dígitos**

**IMPLICIT NONE**

**INTEGER :: n**

**INTEGER, EXTERNAL :: cifras**

**PRINT\*, “introduzca n”**

**READ\*, n**

**Call cifras(n)**

**Print\*, “La respuesta es: ”, cifras**

**END PROGRAM dígitos**

FUNCTION cifras(n)

IMPLICIT NONE

INTEGER,INTENT(IN) :: n

INTEGER ::cifras,i, d

i=0

d=0

DO WHILE(d<=1)

d=n/10

i=i+1

END DO

Cifras=i

END FUNCTION cifras