

LABORATORIO 4

ESTRUCTURAS REPETITIVAS FOR

1 OBJETIVOS

Al finalizar esta actividad, serás capaz de:

- Entender los bucles anidados; Aprenderás las ventajas de utilizar un **for** para aquellos casos en los que sabemos el número de vueltas que vamos a dar.
- Entender la diferencia existente entre utilizar un **while**, un **do-while**, y un **for**.

2 MOTIVACIÓN

La sintaxis del **for** nos resulta mucho más cómoda en algunos casos ya que se escriben la inicialización, la condición de final y las actualizaciones en una misma línea de código.

El for en C

```
for (inicialización; condición; actualización)
{
    Conjunto de instrucciones que han de repetirse
}
```

2.1 Ejercicio de ejemplo

Pedir al usuario un número N y escribir por pantalla el siguiente dibujo

```
*
*
*
```

.... tantas filas como N

Vamos a mostrar 2 soluciones. La primera resuelto con un **while**. La segunda resuelto con un **for**

```
#include <stdio.h>
#include <stdlib.h>


void main()
{
    int n, num;

    printf("\nIntroduce un número entre 1 y 10");
    scanf("%d", &n);

    while ((n < 1) || (n > 10))
    {
        printf("\nEl número no es adecuado.");
        scanf("%d", &n);
    }

    num = 1; /* p.q. en la primera vuelta del while
              queremos imprimir el 1 */

    while (num <= n){
        printf("");
        printf("\n");
        num = num + 1; /* incrementamos el valor de num
                        para que en la siguiente
                        vuelta del while valga uno más */
    } /*final del while,
        es decir final de las acciones a repetir*/
    system("PAUSE");
}
```



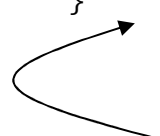
```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int n, num;

    printf("\nIntroduce un número entre 1 y 10");
    scanf("%d", &n);
    while ((n < 1) || (n > 10))
        printf("\nEl número no es adecuado.");
    scanf("%d", &n);
}

for(num = 1; num <= n; num = num + 1){
    printf("");
    printf("\n");
} /*el final del for*/

system("PAUSE");
}
```



3 EJERCICIOS

3.1 Ejercicio 1

3.1.1 Ayuda para el enunciado 1:

Podemos tomar como base el ejercicio ejemplo, pero esta vez como por cada fila no deberemos dibujar sólo una estrella sino varias, deberemos utilizar OTRO bucle adicional para escribir POR CADA LÍNEA tantas estrellas como sea necesario...

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int n, num;

    printf("\nIntroduce un número entre 1 y 10");
    scanf("%d", &n);
    while ((n < 1) || (n > 10))
        printf("\nEl número no es adecuado.");
    scanf("%d", &n);
}

    for(num = 1; num <= n; num = num + 1){
        for (????){
            printf("*");
            .....
        }/*el final del for*/
        .....
    }/*el final del for*/

system("PAUSE");
}
```

3.1.2 Enunciado 1

Pedir por pantalla un número N y dibujar:

```
*
**
***
****    Así hasta N filas.
```

3.2 Ejercicio 2

3.2.1 Ayuda para el enunciado 2:

En este caso no querremos escribir * sino números así que ¿printf("*"); será lo que queramos utilizar?

3.2.2 Enunciado2

Pedir por pantalla un número N y dibujar:

```
1
1 2
1 2 3
1 2 3 4
..... Así hasta N filas.
```

3.3 Ejercicio 3

3.3.1 Objetivo del enunciado 3:

Ver como en algunos casos el **for** no es la instrucción más adecuada y razonarlo

3.3.2 Ayuda para el enunciado 3:

En este caso también nos harán falta dos bucles anidados, pero ¿podrán ser 2 for?

3.3.3 Enunciado

Escribir los factoriales de los números que se van introduciendo por pantalla.

Formato de pantalla:

De que número quieres calcular el factorial? N

N! = XXX

Deseas más factoriales (s/n)?

$N! = n * (n-1) * (n-2) * \dots * 1$

3.4 Ejercicio 4

Calcular la suma de los 10 primeros términos de la siguiente serie:

$$1/2 + 2/2^2 + 3/2^3 + 4/2^4 + \dots + n/2^n$$

Ejemplo:

Vamos a calcular el valor de aplicar la serie $1/2 + 2/4 + 3/8 + 4/16 + \dots$

El resultado de sumar los 10 primeros elementos es 1.988281