



## SESIÓN 3: Estructuras de control: Repetitivas, Iterativas o Bucles

### Objetivos

- Saber construir y utilizar una estructura repetitiva **while**, **for**, **do while**.
- Saber elegir el tipo de bucle adecuado para un problema dado.
- Ser capaces de construir bucles anidados.
- Evitar la construcción de bucles infinitos.

**Nota importante:** Siga el esquema de nombrado de paquetes que se indicó en la sesión 1 es decir: **org.pc.sesion03**. En ese paquete se crearán todos los programas que se proponen en la sesión dándoles un nombre alusivo a lo que realiza el programa y que se indica en cada ejercicio entre paréntesis y en negrita.

Al final de la sesión, el alumno deberá cargar el trabajo realizado a su repositorio indicando la clave correspondiente a la sesión.

### Ejercicios propuestos

1. Haz un programa que muestre los números que se generan con la conjetura de Ulam.

**Nota:** La conjetura de Ulam consiste en lo siguiente: se empieza con cualquier número entero positivo. Si es par se divide entre 2. Si es impar se multiplica por 3 y se le suma 1. Se repite el proceso hasta llegar al número 1. (**ConjeturaUlam**)

Ejemplo de ejecución



```
Introduce un valor entero positivo para generar la sucesión de Ulam
5
Sucesión de Ulam generada a partir de 5
16      8      4      2      1
```

2. Haz un programa que permita obtener el MCD (máximo común divisor) de dos enteros positivos haciendo uso del algoritmo de Euclides. (**Euclides**). Para ello se divide el dividendo (x) entre el divisor (y) y se obtiene un cociente ( $q_1$ ) y un resto ( $r_1$ ) si dicho resto es distinto de 0 se divide nuevamente el anterior divisor entre el resto obtenido. Se continua el proceso hasta obtener un resto 0. El resto anterior a éste es el máximo común divisor de los dos números.

$x \mid y$	$y \mid r_1$	$r_1 \mid r_2$	.....	$r_{n-2} \mid r_{n-1}$
$r_1 \quad q_1$	$r_2 \quad q_2$	$r_3 \quad q_3$		$r_n \quad q_n$
$r_1 \neq 0$	$r_2 \neq 0$	$r_3 \neq 0$	.....	$r_n = 0 \Rightarrow r_{n-1}$ es el mcd

Ejemplo de ejecución



```
Introduce el primer valor entero positivo
-3
Introduce el primer valor entero positivo
25
Introduce el segundo valor entero positivo
10
El MCD de 25 y 10 es 5
```

3. Haz un programa que muestre los K primeros términos de la serie de Fibonacci.

$$f_0 = 0$$

$$f_1 = 1$$

$$f_i = f_{i-1} + f_{i-2} \text{ para } i > 1 \quad \textbf{(Fibonacci)}$$

Ejemplo de ejecución



```
Indica cuántos términos quieres mostrar de la serie de Fibonacci
8
0      1      1      2      3      5      8      13
```

4. Haz un programa que muestre si un número entero positivo ( $>1$ ) es primo. En el archivo EjerciciosResueltos01\_1.pdf, situado en el repositorio de la asignatura en la carpeta Tema01 del GrupoDocente, el ejemplo nº 12 muestra dos maneras de hacerlo y deja planteada una tercera. Resuelve esta última. Prueba la ejecución para números primos y no primos. **(Primo)**

Ejemplo de ejecución



```
Introduce un número (>1) para saber si primo
0
Introduce un número (>1) para saber si primo
9
NO ES PRIMO
```

5. Un entero positivo es un “Número Perfecto” si es igual a la suma de todos sus divisores positivos excluyendo el mismo. Por ejemplo, 6 es el primer número perfecto porque sus divisores (excluyendo el mismo) son 1, 2 y 3 cuya suma  $1 + 2 + 3 = 6$  coincide con el propio número. Haz un programa que pida un número y muestre por consola si es perfecto. **(NumeroPerfecto)**

Ejemplo de ejecución



```
Introduzca un número entero positivo para saber si es perfecto
28
El número es perfecto
```

## Trabajo autónomo

6. Se conoce como Número Armónico a la suma:

$$H_N = 1/1 + 1/2 + 1/3 + \dots + 1/N$$

Haz un programa que pida el valor de N y muestre  $H_N$ . **(Armonico)**

Ejemplo de ejecución



```
Introduce un número para mostrar el Número Armónico
10
El valor del Número Armónico es 2.9289682539682538
```

7. Haz un programa que lea un número indeterminado de enteros y muestre cuantos son positivos, cuantos negativos, número total de valores leídos, suma de los valores positivos, suma de los valores negativos, media de todos los valores leídos (no se contabiliza el 0). El programa termina cuando la entrada es 0, valor centinela. Prueba con una estructura while y con for. **(BucleCentinela)**

Ejemplo de ejecución



```
Introduce valores enteros, el programa termina si la entrada es 0
1 2 -1 3 0
El número de positivos es 3
El número de negativos es 1
El número total de valores leídos es 4
La suma de positivos es 6
La suma de negativos es -1
La media de los valores es 1.25
```

8. Dado un número natural no superior a 20, haz un programa que muestre por consola un triángulo rectángulo con base y altura igual al número y formado por los primeros números naturales escritos en orden inverso. **(TrianguloNumeros)**

Ejemplo de ejecución



```
Introduce un valor no superior a 20
5
TRIÁNGULO GENERADO
15
14      13
12      11      10
9        8        7        6
5        4        3        2        1
```