


<p align="center">Programación</p>	
<p align="center"><i>1º Evaluación</i></p>	
<p align="center">Tema 01. Programación estructurada</p>	

RELACIÓN DE EJERCICIOS 2. Programación estructurada II

Resolver los siguientes problemas en pseudocódigo y en diagrama de flujo:

1. Realizar un programa que solicite 10 números por teclado e indique para cada uno de ellos si es positivo o negativo. El cero se considera positivo.
2. Realizar un programa que lea un número entero entre 0 y 10 y visualice su tabla de multiplicar.
3. Realizar un programa que solicite una cantidad de números que van a introducirse por teclado. Después, para cada uno de ellos que indique si el número es par o impar. Si la cantidad de números introducida es 0 o negativa volverá a solicitarse el dato.
4. Realizar un programa que lea un número estrictamente positivo N y muestre la suma de los N primeros números. Diseña el programa de forma que si N es incorrecto vuelva a solicitarse.

Ejemplo:

```
Introduce un número positivo: 4
La suma de los 4 primeros números es: 10
```

5. Realizar un programa que solicite números hasta que se introduzca un número negativo. Cuando termine informará de cuantos números positivos se han introducido. En este programa el número 0 se considera positivo.

Ejemplo:

```
Introduce un número (negativo para terminar): 4
Introduce un número (negativo para terminar): 8
Introduce un número (negativo para terminar): -2
Has introducido 2 números positivos
```

6. Realizar un programa que lea dos números enteros cualesquiera numeroA y numeroB, y calcule el producto de A y B mediante sumas, es decir, sin usar el operador *.
7. Realizar un programa que solicite una cantidad de números que van a pedirse por teclado. Una vez que solicite todos ellos debe informar de cual es la media de los números. Diseñar el programa de forma que si la cantidad es incorrecta vuelva a solicitarse.
8. Realizar un programa que solicite un conjunto de números. Después de introducir cada número se realizará la pregunta: "¿Desea introducir más números (S/N)". Si la respuesta es 'S' se solicitará otro número. Cuando no desee introducir más números debe informar cual es el menor de los números introducidos.

NOTA: Debe comprobarse que la respuesta es 'S' o 'N' y si no lo es, volver a pedirla.

9. Realizar un programa que lea un número entero positivo y averigüe si es perfecto. Un número es perfecto cuando es igual a la suma de sus divisores excepto él mismo. Diseñar el programa de forma que si algún dato es incorrecto vuelva a solicitarse.

Ejemplo:

El número 6 es perfecto porque es igual que $1+2+3=6$

El número 8 no es perfecto porque $1+2+4=7$

10. Realizar un programa que solicite un número entero positivo o cero y calcule su factorial. Diseñar el programa de forma que si algún dato es incorrecto vuelva a solicitarse.

Explicación:

Factorial (0)=1

Factorial (1)=1

Factorial (N)=N* (N-1) * (N-2) *1