

Este boletín de ejercicios está orientado a practicar con todo lo visto en los boletines anteriores.

1. Escribir un programa que pida un número por teclado y calcule su factorial. Como sabes, la factorial de un número se calcula multiplicando ese número por los sucesivos factores que obtenemos restando uno hasta llegar a la unidad. Por ejemplo, el factorial de 6 (que se representa así 6!) sería este:

$$6! = 6*5*4*3*2*1 = 720$$

2. En matemáticas, la sucesión de Fibonacci se trata de una serie infinita de números naturales. Los dos primeros son siempre el 0 y el 1. Los siguientes se obtienen sumando los dos anteriores. El tercero sería 1 (la suma de 0 + 1), el cuarto sería 2 (la suma de 1 + 1), el quinto 3 (la suma de 1 + 2) y así sucesivamente. La lista con los 10 primeros números sería esta: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34].

Queremos hacer un programa que reciba un número por teclado y nos calcule tantos números de la sucesión de fibonacci como indique ese número. Por ejemplo, si metemos un 8 la salida de tu programa debería de ser así:

0,1,1,2,3,5,8,13

3. Queremos ahora hacer un programa que reciba un número por teclado y nos muestre en orden todos los números de la sucesión de fibonacci que sean menores o iguales al que has enviado como argumento. Por ejemplo, si metemos el número 4 nos debería de devolver esto:

0,1,1,2,3

4. Escribir un programa que cuente el número de cifras que tiene un número (por ejemplo, el 8 tiene una cifra, el 221 tres y el 456789 seis).
5. Escribir un programa que nos diga si un número es capicúa.
6. Escribir un programa que muestre por pantalla los 50 primeros números primos, sus raíces cuadradas, sus cuadrados y sus cubos
7. Decimos que dos números primos son gemelos cuando están separados por un único número (el 11 y el 13, el 17 y el 19, el 41 y el 43, etc.). Escribir un programa que calcule la primera pareja de primos gemelos por encima del 50.
8. Escribe un programa que sume por un lado las cifras pares y por otro las impares de un número y nos muestre ambos resultados. Por ejemplo, si el número en cuestión es el 128 nos debería de decir que la suma de las cifras pares es 9 y la de las impares 2
9. Escribir un programa que nos pida por teclado primero una cadena y luego un carácter. A continuación debe de imprimirnos cuantas veces aparece dicho carácter y en las posiciones de la cadena donde lo hace. Por ejemplo, si nuestra cadena es Hola Mundo y el carácter la o nos debería de decir algo así:

La o aparece en 2 ocasiones

Las posiciones en las que aparece son: 1,9

10. Escribir un programa que nos pida una cadena por teclado y luego nos imprima sólo las cifras que aparecen en ella.

Por ejemplo, si introducimos la cadena "Beverly Hills, 5. CP: 28934" Debería devolvernos: 528934

11. Escribir un programa que nos pida una frase por teclado y luego nos la imprima separando todos los caracteres de sus palabras (excepto los espacios) con un guión. Por ejemplo, si la frase introducida es "esto es una prueba" la salida del programa debería de ser "e-s-t-o e-s u-n-a p-r-u-e-b-a"

12. Crear un programa que lea un número de año por teclado e indique si es bisiesto o no. Un año bisiesto es aquel que es divisible por 4, siempre y cuando no lo sea por 100. La excepción a esta regla son los años múltiplos de 400, que siempre son bisiestos.

13. Hacer un programa que lea un número y un carácter y visualice una matriz compacta repitiendo ese carácter y con tantas filas y columnas como indique el número. Por ejemplo, si metemos el 4 y la x nos debería de mostrar esto:

```
xxxx
xxxx
xxxx
xxxx
```

14. Escribe un programa que lea una hora por teclado en formato 24 horas (HH:MM). Tu programa debería de decir si corresponde a la mañana (entre las 6 y las 11, ambas inclusive), si es una hora de la tarde (entre las 12 y las 19, ambas inclusive), si es de la noche (entre las 20 y las 23, ambas inclusive), si es de la madrugada (entre las 0 y las 5, ambas inclusive) o bien, si el formato no es correcto o no se corresponde con una hora real (minutos de mas de 60, horas negativas o por encima de 23, etc.