

EJERCICIOS – CICLOS CONTROLADOS POR CONTADOR

1. Para todo entero positivo n , el **factorial** de n se define como el producto de todos los números enteros positivos desde 1 hasta n ($n! = 1*2*3*4*...*(n-1)*n$). Se pide crear un programa que pida por pantalla un número y devuelva el factorial de n . El factorial de cero es por definición uno. *Probar dar la salida usando `int` y también “`long long`”.*
2. Mostrar todos los números **pares** comprendidos entre dos variables enteras leídas por teclado.
3. Para hallar los **múltiplos** de un número n , basta multiplicar n por cada uno de los números naturales 0, 1, 2,... Crear un programa que muestre por pantalla los 1000 primeros múltiplos de 5.
4. Realizar un programa para determinar si un entero positivo n es **primo**.
5. Realizar un programa que lea 20 letras ('a' ... 'z') y determine si el número de veces que ha aparecido 'a' es mayor al número de veces de 'z'. *Considerar que puede darse la circunstancia de que alguno de ellos, o ambos, no aparezcan.*
6. Realizar un programa que lea n números reales y muestre por pantalla el real leído más **cercano** al cero.
7. Diseñar un programa que lea n números reales y muestre por pantalla el **segundo** elemento mayor (*el más cercano al mayor dato que es menor a éste*).
8. Desarrollar un programa para mostrar los **divisores** propios de n .
9. Realizar un programa que muestre separadas las cifras de un entero de entrada.
10. Un número se define como **perfecto** si es igual a la suma de sus divisores, excluido él mismo. Por ejemplo, 6 es perfecto, ya que $6 = 1+2+3$. Escribir un programa que pida un número y devuelva por pantalla un mensaje diciendo si es o no perfecto.
11. Determinar si n términos de entrada están **ordenados** de forma creciente.
12. Resolver los ejercicios 1-10 empleando la instrucción **for**.
13. Diseñar un programa en C++ que lea 25 enteros y muestre por pantalla los enteros que existen entre el **mayor de los negativos** leídos y el **menor de los positivos** leídos. *Asumir que el usuario siempre va a introducir números positivos y negativos.*
14. La **Sucesión de Fibonacci** es la siguiente sucesión infinita de números naturales: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377 Esta sucesión se inicia con 1 y 1, y a partir de ahí cada elemento, es la suma de los dos anteriores. Escribir un programa que muestre el término n -ésimo de la *Sucesión de Fibonacci*.
15. De un conjunto de n valores reales de entrada, determinar la **secuencia ordenada de mayor longitud** (*dónde comienza y su longitud*).