

EJERCICIOS DE ESTRUCTURAS DE REPETICIÓN.

Hacer los siguientes algoritmos/programas:

1. Hacer un algoritmo que imprima **N** términos de una suma que comienza con un número **inicial** indicados por el usuario. Si el número **inicial** es el 20 y **N** vale 3, entonces debe imprimirse la suma: **20+21+22= 63**. La suma debe imprimirse completa, tanto términos como resultado de la suma, tal como se mostró anteriormente.
2. En una empresa se desea calcular la productividad de un año, de acuerdo a lo siguiente:

Enero, Febrero y Marzo tienen factor 15
Abril, Mayo y Junio factor 17
Julio y Agosto factor 19
Septiembre, Octubre y Noviembre factor 20
Diciembre factor 21

Escribir un algoritmo que permita calcular la productividad **anual**, si se sabe que ésta se calcula sumando los productos del número de artículos producidos *en cada mes*, multiplicado por el factor que le corresponde.

3. Hacer un algoritmo que solicite un número entero e imprima una a una las cifras que lo componen, comenzando por la última de ellas. Por ejemplo si el número dado fue 14009, el algoritmo debe imprimir **9, 0, 0, 4, 1**
4. Hacer un algoritmo que solicite un número entero e imprima una a una las cifras que lo componen, comenzando por la primera de ellas. Por ejemplo si el número dado fue 60809, el algoritmo debe imprimir **6, 0, 8, 0, 9**
5. Hacer un algoritmo que solicite **N** números y después de ellos se muestre cuál fue el **mayor**, cuál fue el **menor** y el **promedio** de todos. Los ceros dados no se cuentan (se ignoran). **N** es dado por el usuario, así como cada número.
6. Forzar al usuario a ingresar dos valores mayores que cero, usando ciclos. Mostrar el mínimo común divisor (**que no sea 1**) y el máximo común divisor de los dos números dados por el usuario. Por ejemplo, si consideramos los números 18 y 6, el mínimo común divisor de ellos es 2 y el máximo común divisor es 6. Si consideramos el 7 y el 21, el mínimo común divisor de ellos es 7 y el máximo común divisor es también 7. Si consideramos el 13 y el 7, no existe un mínimo común divisor de ellos, ni un máximo común divisor.
7. Hacer un algoritmo que solicite un número e imprima la leyenda **YES!** si la suma de sus cifras (dígitos) pares es igual a la suma de sus cifras impares. Por ejemplo para el número 1430 se imprimiría **YES!**
8. Hacer un algoritmo que calcule y muestre el resultado de multiplicar **N X M**, a través de sumas sucesivas, con el menor número de sumas posibles. Considere que **N** y **M** son enteros ingresados por el usuario y siempre recibirán valores mínimos de cero. (prohibido utilizar el operador *)
9. Hacer un algoritmo que calcule el resultado de dividir **N / M**, a través de restas sucesivas, con el menor número de restas posibles. Considere que **N** y **M** son valores ingresados por el usuario y siempre recibirán valores mínimos de cero. Mostrar cociente y residuo de la división (prohibido utilizar los operadores **/** y **mod**). Considere como error la división entre cero enviando un mensaje que así lo indique.
10. Hacer un algoritmo que calcule e imprima el factorial de **N**. **N** debe ser ingresado por el usuario y siempre será un valor mínimo de 0. El factorial de un número resulta de multiplicar ese número por los valores inferiores a él, hasta llegar a 1. Por ejemplo el factorial de 7 es el resultado de 7x6x5x4x3x2x1. Por definición matemática el factorial de 0 es 1.
11. Hacer un algoritmo que imprima la siguiente sucesión de números:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377...

Observe que la sucesión comienza con los números 0 y 1, y a partir de éstos, los siguientes números se forman con la suma de los dos anteriores. El usuario debe ingresar la cantidad de valores que quiere ver, por ejemplo si el usuario indica que quiere ver 5 números, solo se imprimiría la sucesión 0,1,1,2,3

-
- Figure 1 displays six diagrams (1) through (6) illustrating the evolution of a quantum state in a 2D lattice. The diagrams show the spatial distribution of the state at different stages of time, with asterisks representing the state's presence at specific lattice sites.
- Diagram 1:** Shows a localized state at the top center of the lattice.
 - Diagram 2:** Shows the state spreading horizontally, forming a wider central region.
 - Diagram 3:** Shows the state spreading vertically, forming a taller central region.
 - Diagram 4:** Shows the state spreading horizontally again, forming a wider central region.
 - Diagram 5:** Shows the state spreading vertically again, forming a taller central region.
 - Diagram 6:** Shows the state spreading horizontally again, forming a wider central region.

- Fundamentos de Programación