

Ejercicios Clase de Fundamentos de Programación – CETAV – Mayo 2015

1. Hacer un programa que reciba 10 números ingresados por el usuario y retorne la sumatoria total de esos números
2. Hacer un programa que reciba números ingresados por el usuario en un ciclo. En el momento que reciba un número par imprima "Es par" y concluya el programa
3. Hacer un programa que identifique si el número ingresado por el usuario es primo o no
4. Hacer un programa que reciba 3 números ingresados por el usuario y los almacena en 3 variables separadas. El programa debe retornar el número más alto de los valores ingresados. Para lograr esto es necesario validar los primeros dos números para ver cuál es mayor (usar el símbolo "mayor que": >) y el mayor de estos se compara con la última variable para devolver el número mayor de los 3.
5. Hacer un programa que lea e imprima una serie de números distintos de cero, el algoritmo debe terminar cuando se introduzca el valor Cero y el cero no se debe imprimir. Finalmente también se desea obtener la cantidad de valores leídos.
6. Hacer un programa que reciba dos valores numéricos y un operador matemático, el programa debe hacer el cálculo usando los números en el orden ingresado y el operador matemático. El programa debe imprimir el resultado de la operación.
7. Hacer un programa que reciba 2 números (X & Y) calcule X^Y (X elevado a la Y) e imprima el resultado. Hacer uso de ciclos para multiplicar varias veces (ej: $2^3 = 2 * 2 * 2 = 8$).
8. Hacer un programa que reciba 10 números y devuelva la suma de los números que son pares y la suma de los números que son impares.
9. Hacer un programa que identifique si un número ingresado por el usuario es divisible sin residuo por el divisor también ingresado por el usuario. (Hacer uso de modular).

BONUS: 10. Hacer un programa que imprima los primeros 100 números de la serie Fibonacci. La serie Fibonacci empieza en 1, para calcular el siguiente número de la serie hay que sumar los dos números anteriores. Ejemplo: $1 + 0 = 1$, $1 + 1 = 2$, $1 + 2 = 3$, (...) => 1, 1, 2, 3, 5, 8, 13, (...)