

### Práctica 3 : funciones, programación modular.

1. Construir una función que reciba 3 números enteros y devuelva el promedio (valor real).
2. Construir una función que reciba un valor de tipo entero y que devuelva la letra P si el valor es positivo y la letra N si es cero o negativo. *char*
3. Construir una función que reciba 2 valores reales que representan la base y la altura de un triángulo y retorne la superficie del mismo.
4. Construir una función que reciba un valor entero positivo y devuelva verdadero (true) si es par y falso (false) si no lo es. *bool (función)*
5. Construir una función que calcule el factorial de un número entero mayor o igual que cero.
6. Construir una función que calcule y retorne la edad de una persona en años, la función recibe día, mes y año de la fecha de nacimiento y día, mes y año de la fecha actual.
7. Construir una función lógica que reciba un número de 4 dígitos que corresponde al valor de un año y determine si el año es bisiesto o no. Construir también un programa de prueba.

Nota: año es bisiesto si es divisible por 4 pero no es divisible por 100 ó es divisible por 400

8. En una competencia de salto en largo se presentaron 20 deportistas y cada uno realiza 3 saltos. De cada deportista se ingresa: nombre, día, mes y año de nacimiento y los alcances (en cm.) de los 3 saltos realizados.  
El mejor de los 3 saltos se toma como la marca obtenida por el atleta.  
Construya un algoritmo que:  
a) Para cada atleta informe: nombre, edad y marca obtenida.  
b) Informe el nombre del atleta que obtuvo la mejor marca y el nombre del que obtuvo la peor marca.

Nota 1: para averiguar la edad puede usar la siguiente función. No tiene que desarrollarla.

a) `int edad ( int dia_nac, int mes_nac, int anio_nac);`

Nota2: para obtener la marca debe construir y usar una función.

9. Un entero  $n$  se llama *perfecto* si  $n$  es igual a la suma de todos sus divisores diferentes de él. Por ejemplo, 6 es perfecto porque  $6 = 1+2+3$ .

Construir un algoritmo que ingrese 100 números enteros positivos e informe cuántos de ellos son perfectos.

Nota: para averiguar si un número es perfecto puede usar la función lógica `es_perfecto`.

No tiene que desarrollarla pero debe escribir el prototipo.

10. Un entero  $n$  se llama *abundante* si  $n$  es menor que la suma de sus divisores diferentes de  $n$ . Por ejemplo, 12 es abundante porque  $12 < 1+2+3+4+6$ .

Construir un algoritmo que genere números desde 10 hasta 1000 y muestre aquellos que sean abundantes e impares.

Nota: para averiguar si un número es abundante puede usar la función lógica `es_abundante`.

No tiene que desarrollarla pero debe escribir el prototipo.

11. Ingresar caracteres ( los caracteres son las letras , los dígitos y los símbolos especiales) hasta que ingrese un # que indicará el fin de los datos. Informar qué porcentaje de los caracteres eran dígitos y qué porcentaje eran letras.

Nota: puede utilizar las siguientes funciones C++ :



`int isdigit ( int c );` recibe un carácter y retorna un valor distinto de cero si el carácter es dígito , caso contrario retorna cero

`int isalpha ( int c );` recibe un carácter y retorna un valor distinto de cero si el carácter es alfabético , caso contrario retorna cero

Aviso: si codifica este ejercicio no olvide `#include <cctype.h>`

12. Construir una función que reciba un valor que representa el lado de un cuadrado y retorne la superficie y el perímetro del mismo.
13. Calcular la superficie y el perímetro de 25 cuadrados ingresando el valor del lado de cada uno. Utilice la función anterior.
14. Construir una función que reciba 3 valores enteros y los ordene de menor a mayor. Escribir un programa de prueba.
15. Ingrese las 3 notas de un alumno y muéstre las ordenadas de menor a mayor y de mayor a menor. Utilice la función anterior
16. Construir una función que reciba una fecha como un único número en el formato `aaaammdd` y devuelva el día, el mes y el año por separado.
17. Construir una función (llámela `pow`) que reciba 2 enteros positivos que representan la base y el exponente de una potencia y calcule y muestre el resultado de dicha potencia. Escribir un programa de prueba
18. Ingresar un valor entero positivo `T` que representa la cantidad de jugadores de un equipo de básquet. Ingresar luego el nombre y fecha de nacimiento, en el formato `aaaammdd`, de cada uno de ellos.  
Se solicita que con esta información realice las siguientes tareas:
  - a) Para cada jugador mostrar el nombre y la fecha de nacimiento con en el formato `dd/mm/aaaa`.
  - b) Informar cuántos jugadores nacieron antes de 1995.
19. Construir una función que ingrese y retorne año, mes y día de una fecha. La función debe validar los valores ingresados, por ejemplo que  $0 < \text{año} < 9999$  ;  $1 \leq \text{mes} \leq 12$  ; etc  
 Nota: tenga en cuenta que si el año es bisiesto, el mes de febrero tiene 29 días.
20. En una empresa hay 35 operarios, de cada uno se tiene la siguiente información :
  - a) nombre, sueldo básico, cantidad de horas extra , antigüedad y cantidad de hijos
 Cada mes estos datos son ingresados por un operador; si el operario no tiene antigüedad o hijos el valor ingresado en cada caso es cero, o sea que siempre se ingresan 4 valores por cada operario.  
 El sueldo neto de un operario se obtiene sumando el básico más el plus por salario familiar más el plus por antigüedad y restándole los aportes jubilatorios y de obra social.  
 Se pide mostrar el nombre de cada empleado y el sueldo neto a cobrar e informar también el nombre del operario que tiene mayor cantidad de horas extra en el mes.  
 Nota: considere que existen las siguientes funciones que puede utilizar sin desarrollarlas  
`float plus_ant ( float sue_bas, int antigüedad );`  
   // recibe sueldo básico y antigüedad y retorna el importe del plus por antigüedad o cero si no tiene antigüedad  
`float plus_sal_fam (float sue_bas , int hijos);`  
   // recibe sueldo básico y cantidad de hijos y retorna el importe del plus por salario familiar o cero si no tiene hijos  
`void calcular_mostrar_sueldoneto (float sue_bas, float p_ant, float p_familiar);`  
   // recibe sueldo básico, plus por antigüedad y plus por salario familiar, calcula descuentos y muestra el