

**TRABAJO PRACTICO N° 4** - Resuelva los ejercicios en Dev C++

Parte A: Punteros: son variables que almacenan la dirección de memoria de un objeto.

- A. Intercambio de valores: Crea una función que reciba dos punteros a enteros y los intercambie.
- B. Paso por referencia de un puntero: Crea una función que reciba un puntero a un entero y modifique el valor al que apunta.
- C. Hacer una función que reciba un puntero y compruebe si el número es par o impar, y señalar la posición de memoria donde se está guardando el número.
- D. La función Hogwarts recibirá un puntero a cadena "hechizo", y muestra por pantalla cuantas vocales y consonantes tiene. Testear con 'Expelliarmus', nos muestra: 5 vocales, 7 consonantes.
- E. Crear un procedimiento *amigosDeMario* que contenga un arreglo de personajes {Luigi, Donkey, Yoshi} y reciba un puntero pt a string*, nos muestre los personajes incrementando pt++.

Parte B: Funciones - Una función realiza una tarea concreta y puede ser diseñada, implementada y depurada de manera independiente al resto de código.

4. Realiza una función que tome como parámetros un vector de números y su tamaño, y cambie el signo de los elementos del vector. Hacer otra función que muestra el vector.
5. Escriba una función que reciba un vector y suma sus elementos pares, reutilizar las funciones del punto 4 para cargar y mostrar los vectores.
6. En este ejemplo, tenemos una estructura Abuela que contiene un nombre, edad, y hobby. La función *modificarAbuela* recibe una referencia a una estructura Abuela, y cambia nombre, edad, y hobby (tejer, amasar, bordar). En la función main, creamos una instancia de Abuela llamada *abue1* con su nombre, edad, y hobby inicial. Luego, imprimimos los valores originales. Después, llamamos a la función *modificarAbuela* pasando *abue1* por referencia. Esto significa que los cambios realizados dentro de la función afectarán directamente a la estructura original. Finalmente, imprimir los valores de *abue1* después de la modificación para mostrar que los cambios han sido aplicados.
7. En la función main, llamar a otra función *Friends* que reciba un arreglo de estructuras y registre datos de 6 amigo/as con: nombre, sexo, si es soltero, y su Instagram. Luego un procedimiento *muestra* los datos de los soltera/os. Y una función *medioPomelo*, que carga el *amigoElegido*=" " (paso por referencia) el primer soltero menor a 30 años; debe además, retornar su Instagram.
8. Declare struct Fecha, declare FA(fecha actual) y FC(fecha cumple), luego se compruebe en una función si corresponde al día del cumpleaños. Mostrar un mensaje de felicitaciones de ser así.
9. Realice un programa permita elegir entre contar palabras de una frase y contar la ocurrencia de una palabra en la frase, mientras el usuario lo desee. Para ello valerse de lo siguiente:
 - Una función que permita conocer cuantas palabras hay en una frase recibida como parámetro.
 - Otra función que determine cuantas veces se encuentra una palabra en una frase.
10. Realice un programa que contenga las siguientes funciones o procedimientos según sea:
 - cargaMatriz: que carga una matriz de 4 por 4 con números reales ingresados por el usuario
 - muestraMatriz: nos muestra en pantalla la matriz de 4 por 4 recibida por parámetro
 - intercambioDiagonal: recibe 2 matrices por parámetro e intercambia los valores de las diagonales de ambas.



Escriba un programa que llame a las funciones en el siguiente orden: carga (para las 2 matrices), muestra (para las 2 matrices), intercambia, muestra (para las 2 matrices).

11. Declare un vector *Vec* de tamaño TAM constante. Cree una función *Cuadrado*, a la cual se le envíe *Vec* y lo modifique con el cuadrado de sus valores. Otra función que lo muestre. Reutilice.
 12. Cree una estructura *Persona* que tenga un vector de 30 caracteres, y su edad. Declare una variable de tipo *Persona*, y use una función *pedirDatos* que la cargue. Use *getline* para cargar el nombre. Otra función *mostrarDatos* que reciba la estructura y la muestre. Declare el prototipo de las funciones *pedirDatos* y *mostrarDatos*. Pruébelas desde el menú principal.
 13. Realice una función que dada una matriz y un número de fila de la matriz devuelva el menor de los elementos almacenados en dicha fila.
 14. Escriba una función nombrada *funpot()* que eleve un número entero que se le transmita a una potencia en número entero positivo y despliegue el resultado. El número entero positivo deberá ser el segundo valor transmitido a la función.
 15. Haga un programa en C++ que devuelva la parte fraccionaria de cualquier número introducido por el usuario. Ejemplo, si introducis el número 256.879, debería desplegarse el número 0.879.
 16. Escriba una función llamada *maximo()* que devuelva el valor máximo de tres argumentos que se transmitan a la función cuando sea llamada. Suponga que los 3 argumentos serán del tipo float.
 17. Realice un programa que mientras el usuario lo desee pueda realizar estas operaciones, llamando a la función correspondiente, según las que siguen: **A. Calcula la multiplicación por sumas sucesivas.** Ej. $2 \times 4 = 2 + 2 + 2 + 2$ (suma 4 veces el numero 2 generalizando $n \times m = n + n + \dots (m \text{ veces})$). **B. Calcula la división por restas sucesivas** Ej. $10/2=5$ (5 indica la cantidad de veces que pude restar 2 al 10. Y **C. Calcule la potencia por multiplicaciones sucesivas** Ej $2^3 = 2 \times 2 \times 2$ (multiplica 2 por si mismo 3 veces)
 18. Un número es *primo* si solo es divisible por la unidad y por si mismo. Defina una función que determina si un número es primo, debe recibir el número y retornar verdadero o falso.
 - Realice un programa que, de 2 opciones, **1). Determinar si un número es primo** **2). Calcular todos los números primos menores a un número ingresado por el usuario**. El usuario debe poder realizar estas opciones todas las veces que desee.
 19. Realice una *función* que reciba un número y devuelva el factorial del mismo. El factorial de un número: es el producto del número por todos sus antecesores hasta 1. Ejemplos:
 - Factorial de 0 es 1 (por definición) Se denota $0!=1$
 - Factorial de 1 es 1 Se denota $1!=1$
 - Factorial de 4= $4 \times 3 \times 2 \times 1$ Se denota $4!=24$
 - Factorial de 10= $10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$ Se denota $10!= 3.628.800$
 20. Escriba las siguientes *funciones*:
 - A. Tasa de natalidad:** cantidad de nacimientos en proporción con el total de la población, recibe estos datos como parámetros y devuelve la tasa.
 - B. Tasa de mortalidad:** porcentaje de fallecimientos en relación al total de la población, recibe estos datos como parámetros y devuelve la tasa.
 - C. Población en crecimiento:** devuelve true, si la tasa de natalidad es mayor a la de mortalidad, utilice las funciones descriptas en A y B
 - D. Densidad Poblacional:** habitantes por km²
- Realice un *programa* que permita ingresar, el nombre de un país, su población, su superficie, y luego de la opción de calcular: Densidad poblacional, Tasa de natalidad, Tasa de Mortalidad y Evolución demográfica, mientras el usuario lo desee.