

Guía de ejercicios de repaso en C

Para los siguientes ejercicios los valores con los que se trabajarán se ingresarán desde teclado y se debe imprimir por pantalla el resultado esperado.

1. Calcular el área de un círculo.
2. Convertir grados Celsius a Fahrenheit.
3. Verificar si un número es par o impar.
4. Contar el número de dígitos en un número entero.
5. Revertir un número entero. Por ejemplo: $\text{num} = 123 \rightarrow \text{reverse} = 321$
6. Suma de los primeros n números naturales.
7. Encontrar el máximo entre tres números.
8. Comprobar si un número es primo.
9. Calcular el factorial de un número.

Recordando, el factorial de un número n es el producto de todos los números enteros positivos desde 1 hasta n . Es decir:

$$n! = n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$$

Ejemplo:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

10. Calcular la potencia de un número. Utilizar una función “potencia” y trabajar con float.

Cadena y caracteres

11. Longitud de una cadena. Escribe un programa en C que tome una cadena como entrada del usuario y luego muestre la longitud de esa cadena.
12. Concatenar cadenas. Escribe un programa en C que tome dos cadenas como entrada del usuario y luego concatene esas cadenas en una tercera cadena.
13. Comparar cadenas. Escribe un programa en C que compare dos cadenas y muestre si son iguales o diferentes.
14. Invertir una cadena. Escribe un programa en C que tome una cadena como entrada del usuario y luego muestre la cadena invertida.
15. Contar vocales. Escribe un programa en C que cuente el número de vocales en una cadena dada por el usuario.
16. Eliminar caracteres duplicados. Escribe un programa en C que tome una cadena como entrada del usuario y elimine los caracteres duplicados de esa cadena.
17. Ordenar un arreglo de enteros. Escribe un programa en C que ordene un arreglo de enteros en orden ascendente o descendente. El arreglo debe estar previamente inicializado.

18. Sumar elementos de un arreglo. Escribe un programa en C que sume todos los elementos de un arreglo de enteros previamente inicializado.
19. Encontrar el máximo y mínimo. Escribe un programa en C que encuentre el máximo y el mínimo de un arreglo de enteros previamente inicializado.
20. Buscar un elemento en un arreglo. Escribe un programa en C que busque un elemento dado por el usuario en un arreglo de enteros y muestre su posición si se encuentra. El arreglo debe estar previamente inicializado.

Estructuras

21. Punto en el plano. Define una estructura llamada **Punto** que represente un punto en el plano cartesiano con coordenadas x e y. Luego, escribe una función que calcule la distancia (empleando Pitágoras).
22. Ejercicio de Rectángulo. Define una estructura llamada **Rectangulo** que represente un rectángulo con longitud y ancho. Escribe funciones para calcular el área y el perímetro del rectángulo.
23. Ejercicio de Triángulo. Define una estructura llamada **Triangulo** que represente un triángulo con base y altura. Escribe funciones para calcular el área y perímetro. Combina con el ejercicio del Rectángulo y crea dos funciones que permita sumar el área y perímetro de ambas estructuras.
24. Ejercicio de Tiempo. Define una estructura llamada **Tiempo** que represente una hora (horas, minutos y segundos). Escribe funciones para sumar y restar tiempos. Se deberá imprimir en pantalla el resultado en formato HH:MM:SS.
25. Ejercicio de Ecuaciones Cuadráticas. Define una estructura llamada **EcuacionCuadratica** que represente una ecuación cuadrática en la forma $ax^2+bx+c = 0$. Implementa un programa que permita al usuario ingresar los coeficientes de una ecuación cuadrática y luego calcule y muestre las raíces de la ecuación. El programa debe manejar correctamente el caso de raíces reales, imaginarias y repetidas.

Uniones

26. Realice un programa que permita transformar dos datos de 16 bits en uno de 32 bits empleando uniones. Uno de los datos de 16 bits será la parte alta y el otro la parte baja del dato de 32 bits. Se debe ingresar por teclado la parte alta y baja para luego imprimir el resultado en 32 bits.
27. Desarrolla un programa para un sistema de visualización de datos meteorológicos. El programa debe permitir al usuario ingresar la temperatura actual en grados Celsius y luego elegir la unidad en la que desea visualizarla (Celsius o Fahrenheit). De acuerdo con la elección del usuario, el programa debe mostrar la temperatura en la unidad correspondiente. Para ahorrar memoria se debe almacenar la temperatura (Celsius y Fahrenheit) en una unión.