

1. Uso de la instrucción break en un bucle:

Crea un programa que recorra un array de números e imprima cada número hasta que encuentre un valor mayor que 50, momento en el cual el bucle debe detenerse usando la instrucción break.

2. Instrucción continue en un bucle:

Diseña un programa que recorra los números del 1 al 20 e imprima solo los números impares, utilizando la instrucción continue para saltar los números pares.

3. Uso de una instrucción labeled:

Desarrolla un programa que contenga dos bucles anidados. Si se encuentra un valor mayor que 10 en el bucle interno, el programa debe salir del bucle etiquetado y mostrar un mensaje indicando el valor encontrado.

4. Operadores de comparación:

Crea un programa que compare dos números ingresados por el usuario y determine cuál es mayor, cuál es menor o si ambos son iguales, mostrando el resultado con operadores de comparación (==, >, <).

5. Operadores aritméticos:

Diseña un programa que pida al usuario dos números y realice las operaciones de suma, resta, multiplicación, división y módulo. Muestra el resultado de cada operación.

6. Operadores de asignación:

Implementa un programa que declare una variable con un valor inicial y luego utilice operadores de asignación (+=, -=, *=, /=) para modificar el valor de la variable, mostrando el resultado en cada paso.

7. Operadores booleanos:

Desarrolla un programa que utilice operadores lógicos (&&, ||, !) para verificar si un usuario ha ingresado una edad válida (entre 18 y 65) y una contraseña correcta (al menos 8 caracteres). Muestra si ambas condiciones son verdaderas o no.

8. Operadores bit a bit:

Crea un programa que tome dos números e imprima el resultado de las operaciones bit a bit AND (&), OR (|), XOR (^) y NOT (~), explicando el resultado de cada operación.

9. Operadores de objetos:

Implementa un programa que declare un objeto de persona con propiedades como nombre, edad y ocupación. Luego, utiliza el operador delete para eliminar una propiedad del objeto y muestra el objeto antes y después de la operación.

10. Precedencia de operadores:

Diseña un programa que realice una operación matemática compleja, utilizando operadores aritméticos y booleanos, donde la precedencia de los operadores influya en el resultado final. Muestra cómo cambiaría el resultado si se alterara la precedencia con paréntesis.