

Universidad Mariano Gálvez De Guatemala

Facultad de ingeniería, Matemática y ciencias Físicas

Campus Villa Nueva Guatemala

Ingeniería en Sistemas de Información y ciencias de la computación

Curso: programación 1

Código de curso: 012

Código de la carrera: 5090

Estudiante: Coosemans Spiegeler Max Eduardo

Sección: A

Numero de carné: 5090-23-3282

Fecha: 02/02/2024



## introducción

Las expresiones permiten combinar datos y operadores para calcular otros datos, es decir, obtener un resultado.

Los datos pueden ser variables o constantes literales.

Un operador es un símbolo que determina la operación a realizar sobre los operandos (datos) a los que afecta.

Además, pueden contener palabras clave del lenguaje y signos de puntuación.

Una sentencia es un conjunto de expresiones que permiten ejecutar una determinada acción.

En C++, las sentencias simples se caracterizan por terminar con el signo de puntuación;

Entrada y salida estándar (E/S): Se utilizan las funciones `cin` y `cout` para la entrada y salida de datos desde y hacia la consola respectivamente. Estas pertenecen a la biblioteca estándar de C++.

Funciones: Se definen funciones para encapsular bloques de código que realizan tareas específicas. Esto promueve la modularidad y la reutilización del código.

Condicionales (if-else): Se utilizan para tomar decisiones basadas en ciertas condiciones. Por ejemplo, para verificar si un número es primo, determinar si un año es bisiesto o seleccionar una opción del menú.

Bucles (for, do-while): Se utilizan para repetir bloques de código varias veces. En los ejemplos, se emplea un bucle `do-while` para permitir que el usuario realice múltiples operaciones sin reiniciar el programa.

Switch-case: Se usa para seleccionar una opción específica de un conjunto de opciones. En el menú interactivo, se utiliza `switch-case` para manejar la selección del usuario.

Funciones matemáticas básicas: Se utilizan operaciones aritméticas básicas como suma, resta, multiplicación y división en algunos programas.

Tipos de datos: Se utilizan tipos de datos como `int`, `double` y `char` para representar números enteros, números de punto flotante y caracteres respectivamente.

Operadores lógicos y aritméticos: Se utilizan operadores lógicos (`&&`, `||`, `!`) y aritméticos (`+`, `-`, `*`, `/`, `%`) para realizar operaciones lógicas y aritméticas en las condiciones y expresiones.

Funciones booleanas: Se definen funciones que devuelven valores booleanos para indicar si una condición es verdadera o falsa, como determinar si un número es primo o si un año es bisiesto.

Algoritmos básicos: Se implementan algoritmos básicos para calcular potencias, verificar si un número es primo y determinar si un año es bisiesto.

**Modularidad y reutilización del código:** El modularidad se logra mediante la definición de funciones que realizan tareas específicas, lo que facilita la comprensión y mantenimiento del código. Además, la reutilización del código se promueve al encapsular funcionalidades en funciones que pueden ser llamadas desde diferentes partes del programa.

**Interactividad con el usuario:** Los programas desarrollados interactúan con el usuario a través de la entrada y salida estándar (E/S). Esto permite al usuario ingresar datos y recibir resultados de manera amigable y comprensible.

**Uso de estructuras de control:** Se utilizan diversas estructuras de control, como condicionales (if-else), bucles (for, do-while) y switch-case, para controlar el flujo del programa y tomar decisiones basadas en condiciones específicas.

**Manipulación de tipos de datos y operaciones:** Se manipulan varios tipos de datos, como enteros, flotantes y caracteres, y se realizan operaciones aritméticas y lógicas con ellos. Esto demuestra la importancia de comprender y utilizar los tipos de datos y operadores proporcionados por el lenguaje de programación.

**Principios algorítmicos básicos:** Se implementan algoritmos básicos para calcular potencias, verificar si un número es primo y determinar si un año es bisiesto. Estos ejercicios ayudan a fortalecer la comprensión de algoritmos simples y su implementación en C++.

1. Operadores y Variables:

```
Output
/tmp/Gb6iiaDwT.o
Ingrese el primer numero entero: 78
Ingrese el segundo numero entero: 99
Resultados:
Suma: 177
Resta: -21
Multiplicacion: 7722
Division: 0.787879
Comparaciones:
Los numeros son iguales: 0
El primer numero es mayor que el segundo: 0
El primer numero es menor que el segundo: 1
```

2. Funciones:

```
Output Clear
/tmp/Q0BwCR9is9.o
Ingrese la base: 78
Ingrese el exponente: 2
El resultado de 78 elevado a 2 es: 6084
```

3. Determinar Números Primos:

```
Output
/tmp/g47jEGJQqd.o
Ingrese un numero: 8
8 no es un numero primo.
```

4. Año Bisiesto

```
Output
/tmp/rZZTYfkjMv.o
Ingrese un año: 1495
1495 no es un año bisiesto.
```

## 5. Menú de Opciones:

```
Output Clear
/tmp/gJxk0PFyup.o
Seleccione una opcion:
a. Calcular potencia
b. Verificar si un numero es primo
c. Determinar si un año es bisiestro
x. Salir
Opcion: a
Ingrese la base: 78
Ingrese el exponente: 9
El resultado es: 1.06869e+17
Seleccione una opcion:
a. Calcular potencia
b. Verificar si un numero es primo
c. Determinar si un año es bisiestro
x. Salir
Opcion: b
Ingrese un numero: 45
El numero 45 no es primo.
Seleccione una opcion:
a. Calcular potencia
b. Verificar si un numero es primo
c. Determinar si un año es bisiestro
x. Salir
Opcion: c
Ingrese un año: 8965
8965 no es un año bisiestro.
Seleccione una opcion:
a. Calcular potencia
b. Verificar si un numero es primo
c. Determinar si un año es bisiestro
x. Salir
Opcion: x
Saliendo del programa...
```

1.operadores y Variables:

```
#include <iostream>
using namespace std;

int main() {
    int num1, num2;
    int suma, resta, multiplicacion;
    double division;
    bool igual, mayor, menor;

    // Solicitar al usuario que ingrese dos números enteros
    cout << "Ingrese el primer numero entero: ";
    cin >> num1;

    cout << "Ingrese el segundo numero entero: ";
    cin >> num2;

    // Realizar operaciones aritméticas
    suma = num1 + num2;
    resta = num1 - num2;
    multiplicacion = num1 * num2;

    // Verificar la división por cero antes de realizarla
    if (num2 != 0) {
        division = static_cast<double>(num1) / num2;
    } else {
        cout << "No se puede realizar la division. El segundo numero es cero." << endl;
        division = 0; // Asignar un valor por defecto
    }

    // Realizar comparaciones usando operadores de relación
    igual = (num1 == num2);
    mayor = (num1 > num2);
    menor = (num1 < num2);

    // Mostrar los resultados al usuario
    cout << "Resultados:" << endl;
    cout << "Suma: " << suma << endl;
    cout << "Resta: " << resta << endl;
    cout << "Multiplicacion: " << multiplicacion << endl;
    cout << "Division: " << division << endl;
    cout << "Comparaciones:" << endl;
    cout << "Los numeros son iguales: " << igual << endl;
    cout << "El primer numero es mayor que el segundo: " << mayor << endl;
    cout << "El primer numero es menor que el segundo: " << menor << endl;

    return 0;
}
```

2.Funciones:

```
#include <iostream>
```

```
using namespace std;
```

```
// Definición de la función calcularPotencia
```

```
double calcularPotencia(double base, int exponente) {
```

```
    // Caso base: exponente es cero, el resultado siempre es 1
```

```
    if (exponente == 0) {
```

```
        return 1.0;
```

```
    }
```

```
    // Caso base: exponente es negativo
```

```
    else if (exponente < 0) {
```

```
        // Invertir la base y el exponente para calcular la potencia inversa
```

```
        base = 1 / base;
```

```
        exponente = -exponente;
```

```
    }
```

```
    double resultado = 1.0;
```

```
    // Calcular la potencia mediante multiplicaciones sucesivas
```

```
    for (int i = 0; i < exponente; ++i) {
```

```
        resultado *= base;
```

```
    }
```

```
    return resultado;
```

```
}
```

```
int main() {
```

```
    double base;
```

```
    int exponente;
```

```
    // Solicitar al usuario que ingrese la base y el exponente
```

```
    cout << "Ingrese la base: ";
```

```
    cin >> base;
```

```
    cout << "Ingrese el exponente: ";
```

```
    cin >> exponente;
```

```
    // Llamar a la función calcularPotencia y mostrar el resultado
```

```
    double resultado = calcularPotencia(base, exponente);
```

```
    cout << "El resultado de " << base << " elevado a " << exponente << " es: " << resultado  
<< endl;
```

```
    return 0;
```

```
}
```



### 3. Determinar Números Primos:

```
#include <iostream>
using namespace std;

// Función para verificar si un número es primo
bool esPrimo(int numero) {
    // Los números menores o iguales a 1 no son primos
    if (numero <= 1) {
        return false;
    }

    // Verificar si el número es divisible por algún otro número menor que él mismo
    for (int i = 2; i <= numero / 2; ++i) {
        if (numero % i == 0) {
            return false; // No es primo si es divisible
        }
    }
    return true; // Es primo si no es divisible por ningún otro número
}

int main() {
    int numero;

    // Solicitar al usuario que ingrese un número
    cout << "Ingrese un numero: ";
    cin >> numero;

    // Llamar a la función esPrimo para determinar si el número es primo o no
    if (esPrimo(numero)) {
        cout << numero << " es un numero primo." << endl;
    } else {
        cout << numero << " no es un numero primo." << endl;
    }

    return 0;
}
```

#### 4. Año Bisiesto

```
#include <iostream>
```

```
using namespace std;
```

```
// Función para determinar si un año es bisiesto
```

```
bool esBisiesto(int anio) {
```

```
    if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
```

```
        return true;
```

```
    } else {
```

```
        return false;
```

```
    }
```

```
}
```

```
int main() {
```

```
    int anio;
```

```
    // Solicitar al usuario que ingrese un año
```

```
    cout << "Ingrese un año: ";
```

```
    cin >> anio;
```

```
    // Utilizar la función esBisiesto para determinar si es bisiesto o no
```

```
    if (esBisiesto(anio)) {
```

```
        cout << anio << " es un año bisiesto." << endl;
```

```
    } else {
```

```
        cout << anio << " no es un año bisiesto." << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

## 5. Menú de Opciones:

```
#include <iostream>
```

```
using namespace std;
```

```
// Función para calcular la potencia
```

```
double calcularPotencia(double base, int exponente) {
```

```
    if (exponente == 0) {
```

```
        return 1.0;
```

```
    } else if (exponente < 0) {
```

```
        base = 1 / base;
```

```
        exponente = -exponente;
```

```
    }
```

```
    double resultado = 1.0;
```

```
    for (int i = 0; i < exponente; ++i) {
```

```
        resultado *= base;
```

```
    }
```

```
    return resultado;
```

```
}
```

```
// Función para verificar si un número es primo
```

```
bool esPrimo(int numero) {
```

```
    if (numero <= 1) {
```

```
        return false;
```

```
    }
```

```
    for (int i = 2; i <= numero / 2; ++i) {
```

```
        if (numero % i == 0) {
```

```
            return false;
```

```
        }
```

```

    }

    return true;
}

// Función para determinar si un año es bisiesto
bool esBisiesto(int anio) {
    if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
        return true;
    } else {
        return false;
    }
}

```

```

int main() {
    char opcion;
    double base;
    int exponente, numero, anio;

    do {
        cout << "Seleccione una opcion:" << endl;
        cout << "a. Calcular potencia" << endl;
        cout << "b. Verificar si un numero es primo" << endl;
        cout << "c. Determinar si un año es bisiesto" << endl;
        cout << "x. Salir" << endl;
        cout << "Opcion: ";
        cin >> opcion;

        switch (opcion) {
            case 'a':

```

```
    cout << "Ingrese la base: ";

    cin >> base;

    cout << "Ingrese el exponente: ";

    cin >> exponente;

    cout << "El resultado es: " << calcularPotencia(base, exponente) << endl;

    break;

case 'b':

    cout << "Ingrese un numero: ";

    cin >> numero;

    if (esPrimo(numero)) {

        cout << "El numero " << numero << " es primo." << endl;

    } else {

        cout << "El numero " << numero << " no es primo." << endl;

    }

    break;

case 'c':

    cout << "Ingrese un año: ";

    cin >> anio;

    if (esBisiesto(anio)) {

        cout << anio << " es un año bisiesto." << endl;

    } else {

        cout << anio << " no es un año bisiesto." << endl;

    }

    break;

case 'x':

    cout << "Saliendo del programa..." << endl;

    break;

default:

    cout << "Opcion no valida. Intente de nuevo." << endl;
```

```
    }  
    } while (opcion != 'x');  
  
    return 0;  
}
```