

Universidad Mariano Gálvez De Guatemala
Faculta De Ingeniería Matemáticas Y Ciencias Físicas
Campus Villa Nueva
Ingeniería en sistemas
Código carrera: 5090
Curso: Programación I
Código de curso: 5090-012
Sección A
Docente: Msc. Ing. Carlos Alejandro Arias



"Laboratorio de Programación I: Introducción a Funciones y Operadores en C++"

Introducción

El siguiente trabajo tiene como objetivo presentar programas que implementan del lenguaje de programación C++, dichos programas abordan conceptos fundamentales de operaciones que destacan en el lenguaje, lo cual permite poner en practicar operaciones lógicas, aritmética, introduciéndonos al desarrollo de programas con funciones en el algún inciso los cuales solicitan programas de determinación y verificación, esto permitirá que el código este mas ordenado.

A continuación, podrá observar el código del programa con detalles sobre su funcionalidad a través de comentarios detallados que proporcionan una comprensión clara de cómo utilizarlos y comprender el programa, adjunta ejemplos de ejecución, así como instrucciones detalladas sobre las variables de entrada necesarias para una compilación correcta.

Este informe pretende ser una guía completa para aquellos que deseen comprender y utilizar el programa implementado, destacando tanto su funcionalidad como su estructura interna.

1. Operadores y Variables:

El siguiente programa tiene como objetivo realizar operaciones aritméticas básicas (Suma, Resta, multiplicación y División) y operaciones de relación ($=$, $!=$, $>$, $<$, $>=$, $<=$) que permitirá realizar comparaciones entre nuestras variables.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
//-----  
-----
```

```
//Declarion de variables
```

```
    int num1, num2;
```

```
        // Solucitud de variables
```

```
    cout << "Ingrese el primer numero entero: ";
```

```
    cin >> num1;
```

```
    cout << "Ingrese el segundo numero entero: ";
```

```
    cin >> num2;
```

```
//-----  
-----
```

```
//Declarion de variables
```

```
    // Operaciones aritméticas básicas
```

```
    int sum = num1 + num2;
```

```
    int rest = num1 - num2;
```

```
    int mult = num1 * num2;
```

```

// Division con condicion en 0

float div;

if (num2 != 0) {
    div = static_cast<float>(num1) / num2; //Generador de divisiones con
resultados decimales
} else {
    cout << "No se puede dividir ente cero, ERROR." << endl;
    return 1; // El programa se simplemente retorna.
}

// Operadores de relacion ==, !=,>,<,>=,<=

bool igual = num1 == num2;
bool dif = num1 != num2;
bool mayor = num1 > num2;
bool menor = num1 < num2;

// Impresion de resultados en pantalla
cout << "\nRESULTADOS:" << endl;
    cout << "Suma: " << sum << endl;
cout << "Resta: " << rest << endl;
cout << "Multiplicacion: " << mult << endl;
cout << "Division: " << div << endl;

cout << "\nCOMPARACIONES ENTRE VARIABLES:" << std::endl;
    cout << num1 << " es igual a " << num2 << "? " << boolalpha << igual <<
endl;

    cout << num1 << " es diferente de " << num2 << "? " << boolalpha << dif
<< endl;

```

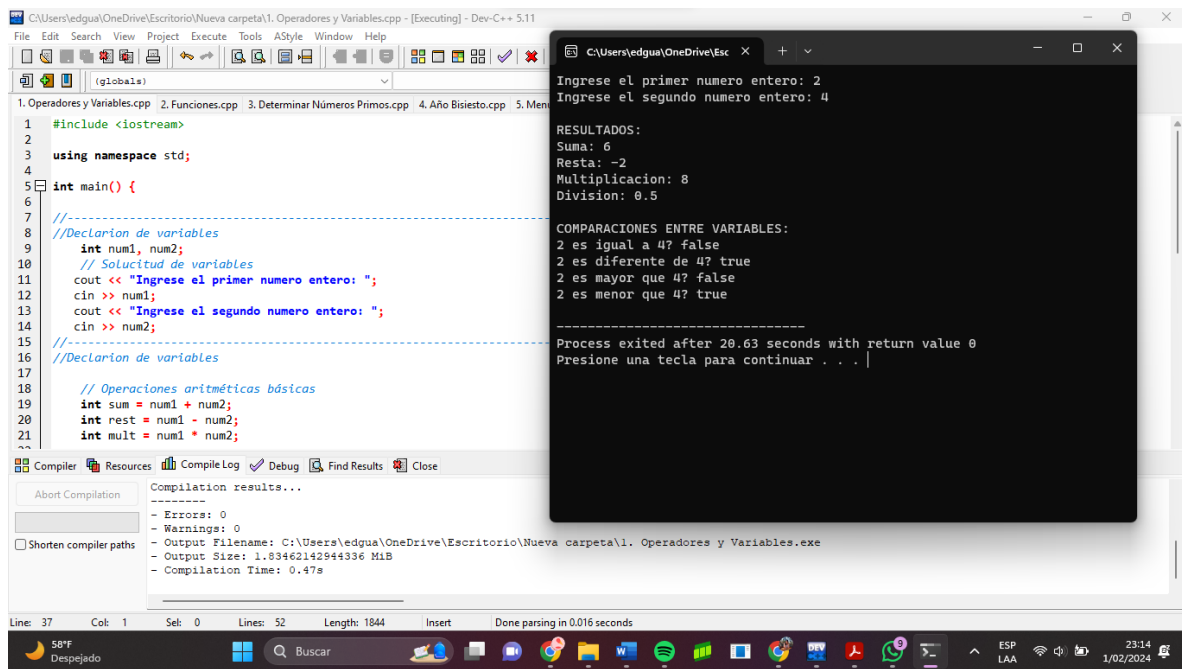
```
cout << num1 << " es mayor que " << num2 << "? " << boolalpha <<
mayor << endl;
```

```
cout << num1 << " es menor que " << num2 << "? " << boolalpha <<
menor << endl;
```

```
return 0; // Final del programa
```

```
}
```

Valores de entrada sugeridos para una ejecución correcta: números enteros o números naturales: 1, 2, 3, 4, 5, 10, 125, 590, 1926, 76409, 9.483.920, junto con cada número negativo correspondiente: -1,-2, -3, -4, -5,-10, -590, -1926, -76409, -9.483.920. Esto incluye, claro, al cero (0).



The screenshot shows the Dev-C++ IDE with a C++ program being executed. The program performs arithmetic operations and comparisons on two input numbers, 2 and 4. The output window displays the results of these operations and comparisons.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6
7     //Declaracion de variables
8     int num1, num2;
9     //Solicitud de variables
10    cout << "Ingrese el primer numero entero: ";
11    cin >> num1;
12    cout << "Ingrese el segundo numero entero: ";
13    cin >> num2;
14
15    //Declaracion de variables
16
17    // Operaciones aritméticas básicas
18    int sum = num1 + num2;
19    int rest = num1 - num2;
20    int mult = num1 * num2;
```

Output window content:

```
Ingrese el primer numero entero: 2
Ingrese el segundo numero entero: 4

RESULTADOS:
Suma: 6
Resta: -2
Multiplicacion: 8
Division: 0.5

COMPARACIONES ENTRE VARIABLES:
2 es igual a 4? false
2 es diferente de 4? true
2 es mayor que 4? false
2 es menor que 4? true

Process exited after 20.63 seconds with return value 0
Presione una tecla para continuar . . .
```

Compiler output:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\edgua\OneDrive\Escritorio\Nueva carpeta\1. Operadores y Variables.exe
- Output Size: 1.83462142944336 KiB
- Compilation Time: 0.47s
```

C:\Users\edgua\OneDrive\Escritorio\Nueva carpeta\1. Operadores y Variables.cpp - [Executing] - Dev-C++ 5.11

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6
7     //Declarion de variables
8     int num1, num2;
9     // Solicitud de variables
10    cout << "Ingrese el primer numero entero: ";
11    cin >> num1;
12    cout << "Ingrese el segundo numero entero: ";
13    cin >> num2;
14
15    //Declarion de variables
16
17    // Operaciones aritméticas básicas
18    int sum = num1 + num2;
19    int rest = num1 - num2;
20    int mult = num1 * num2;
21
22 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\edgua\OneDrive\Escritorio\Nueva carpeta\1. Operadores y Variables.exe
- Output Size: 1.83462142949336 MiB
- Compilation Time: 0.42s

Line: 37 Col: 1 Sel: 0 Lines: 52 Length: 1844 Insert Done parsing in 0.016 seconds

50°F Despejado

23:15 1/02/2024

C:\Users\edgua\OneDrive\Esc X + -

Ingrese el primer numero entero: 10
Ingrese el segundo numero entero: 3

RESULTADOS:
Suma: 13
Resta: 7
Multiplicacion: 30
Division: 3.33333

COMPARACIONES ENTRE VARIABLES:
10 es igual a 3? false
10 es diferente de 3? true
10 es mayor que 3? true
10 es menor que 3? false

Process exited after 17.96 seconds with return value 0
Presione una tecla para continuar . . . |

C:\Users\edgua\OneDrive\Escritorio\Nueva carpeta\1. Operadores y Variables.cpp - [Executing] - Dev-C++ 5.11

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6
7     //Declarion de variables
8     int num1, num2;
9     // Solicitud de variables
10    cout << "Ingrese el primer numero entero: ";
11    cin >> num1;
12    cout << "Ingrese el segundo numero entero: ";
13    cin >> num2;
14
15    //Declarion de variables
16
17    // Operaciones aritméticas básicas
18    int sum = num1 + num2;
19    int rest = num1 - num2;
20    int mult = num1 * num2;
21
22 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\edgua\OneDrive\Escritorio\Nueva carpeta\1. Operadores y Variables.exe
- Output Size: 1.83462142949336 MiB
- Compilation Time: 0.42s

Line: 37 Col: 1 Sel: 0 Lines: 52 Length: 1844 Insert Done parsing in 0.016 seconds

50°F Despejado

23:17 1/02/2024

C:\Users\edgua\OneDrive\Esc X + -

Ingrese el primer numero entero: 0
Ingrese el segundo numero entero: 1

RESULTADOS:
Suma: 1
Resta: -1
Multiplicacion: 0
Division: 0

COMPARACIONES ENTRE VARIABLES:
0 es igual a 1? false
0 es diferente de 1? true
0 es mayor que 1? false
0 es menor que 1? true

Process exited after 4.493 seconds with return value 0
Presione una tecla para continuar . . . |

2. Funciones:

El siguiente programa tiene como objetivo devolver resultados de elevación de la base al exponente.

```
#include <iostream>
```

```
#include <cmath> // Biblioteca para la función pow()
```

```
using namespace std;
```

```
//-----  
-----
```

```
// Funcion para calcular la potencia
```

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

```
    // Condicion en de exponente cero
```

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

```
        return 1.0;
```

```
    }
```

```
    // Condicion en caso de exponente negativo
```

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

```
        return 1.0 / pow(base, -exponente);
```

```
    }
```

```
    // Potencia normal (sin exponente negativo o cero)
```

```
    else {
```

```
        return pow(base, exponente);
```

```
    }
```

```
}
```

```
//-----  
-----
```

```

int main() {

    //Declarion de variables

    float base;

    int exponente;

        // Solicitud de variables

    cout << "Ingrese la base: ";

    cin >> base;


    cout << "Ingrese el exponente: ";

    cin >> exponente;


    // Llamar a la funcion calcularPotencia

    float resultado = calcularPotencia(base, exponente);

        // Mostrando resultado de la funcion calcularPotencia

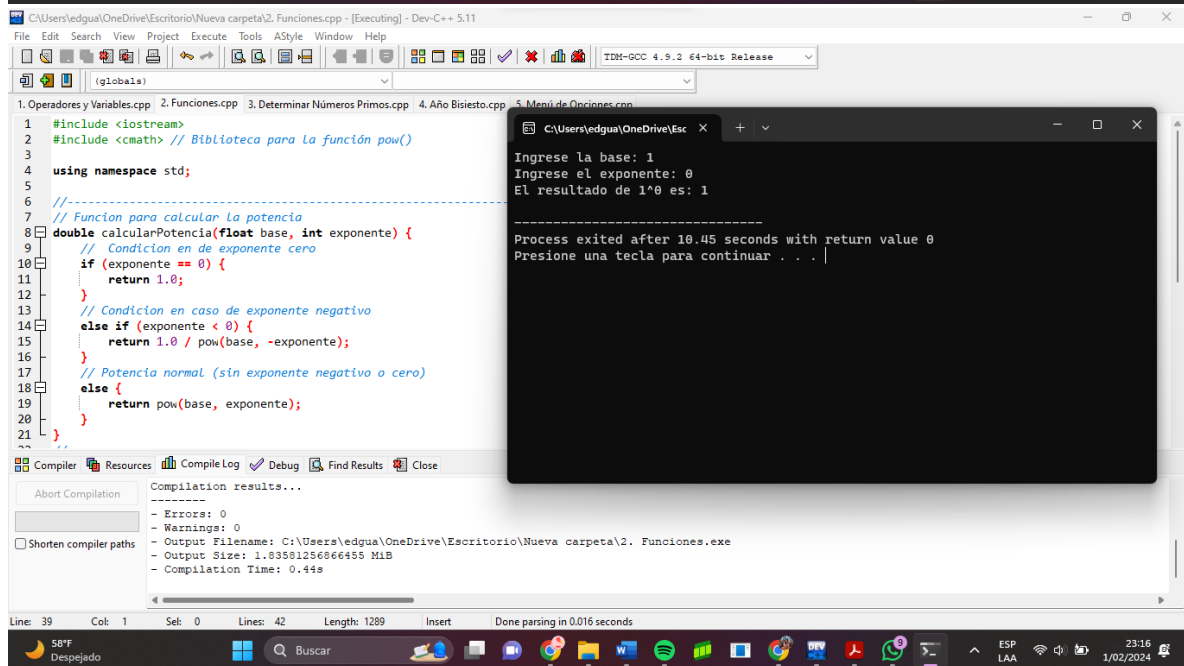
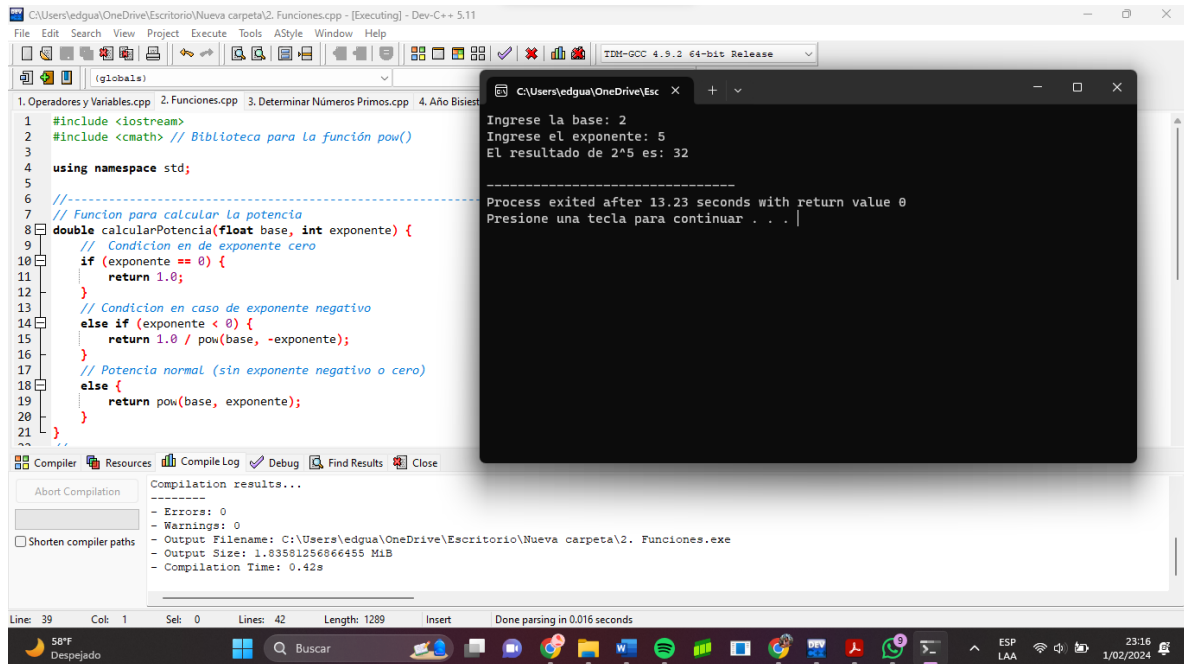
    cout << "El resultado de " << base << "^" << exponente << " es: " <<
resultado << endl;

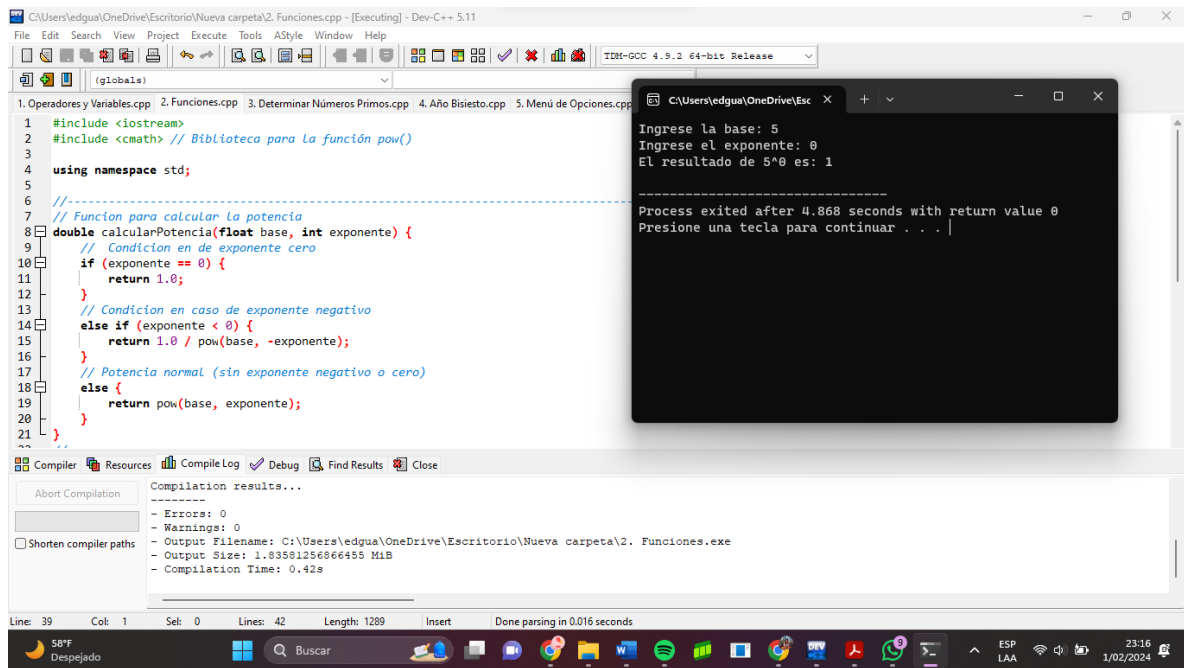

    return 0; // Final del programa

}

```

Valores de entrada sugeridos para una ejecución correcta: utiliza los diez dígitos: 0 (cero), 1 (uno), 2 (dos), 3 (tres), 4 (cuatro), 5 (cinco), 6 (seis), 7 (siete), 8 (ocho) y 9 (nueve) solos o para formar números mas grandes.





3. Determinar Números Primos:

El siguiente programa tiene como objetivo determinar si un número es primo.

```
#include <iostream>
```

```
using namespace std;
```

```
//-----  
-----
```

```
// Funcion para verificar si un numero es primo
```

```
// La funcion retornara un valor Booleano (en caso de ser primo true/ en  
caso de no ser primo false)
```

```
bool esPrimo(int num) {
```

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

```
        return false; // 0 y 1 no son primos po lo tanto la condicion los incluye y  
retorna un valor falso
```

```
    }
```

```
    // Procedimiento para comprobar la divisibilidad por numeros desde 2  
hasta la raíz cuadrada de num
```

```
    for (int i = 2; i * i <= num; ++i) {
```

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

```
            return false; // No es primo, ya que es divisible por i
```

```
        }
```

```
    }
```

```
    return true; // Es primo
```

```

}

//-----

int main() {
    //Declarion de variable
    int num;

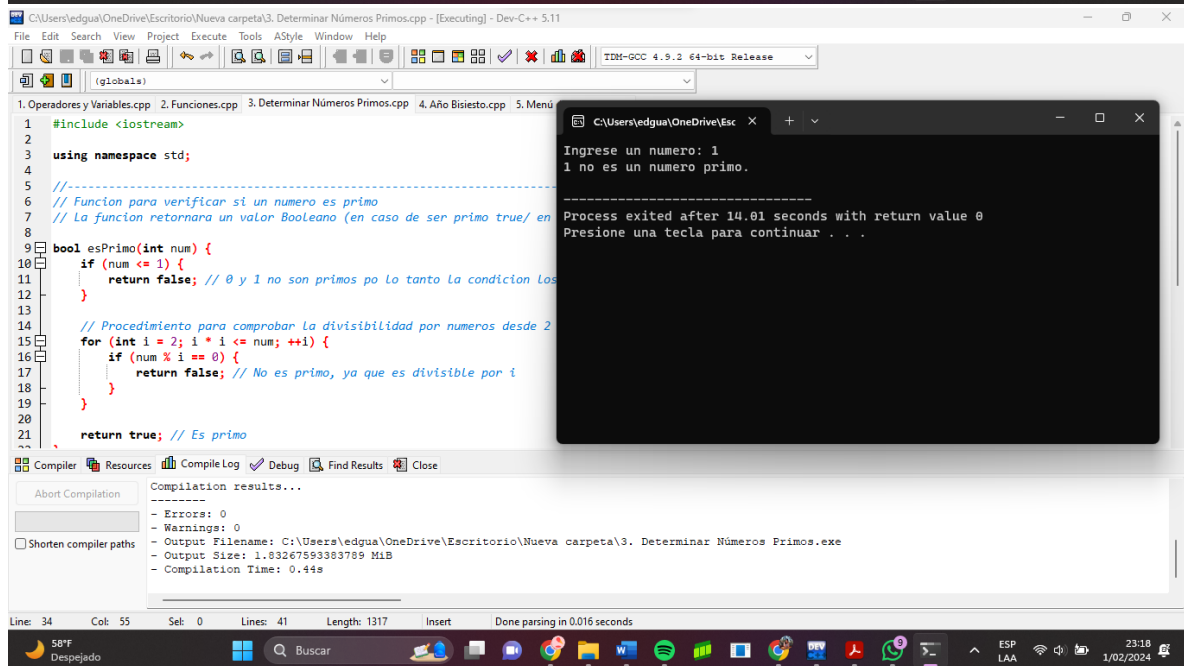
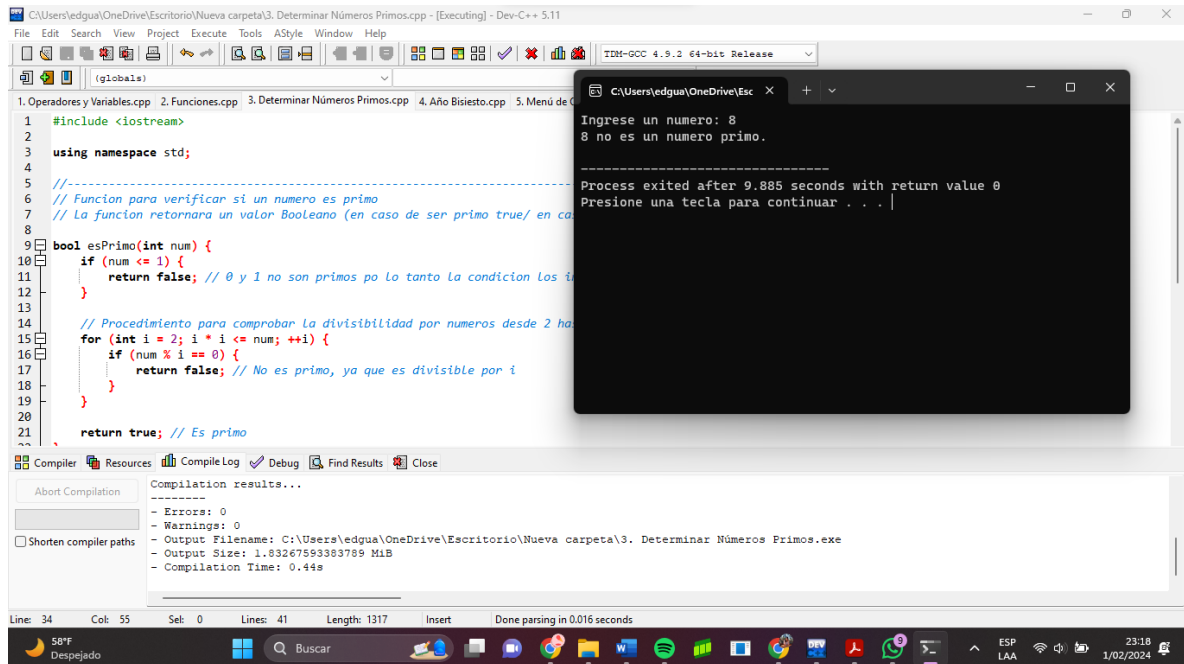
    // Solicitud de variable
    cout << "Ingrese un numero: ";
    cin >> num;

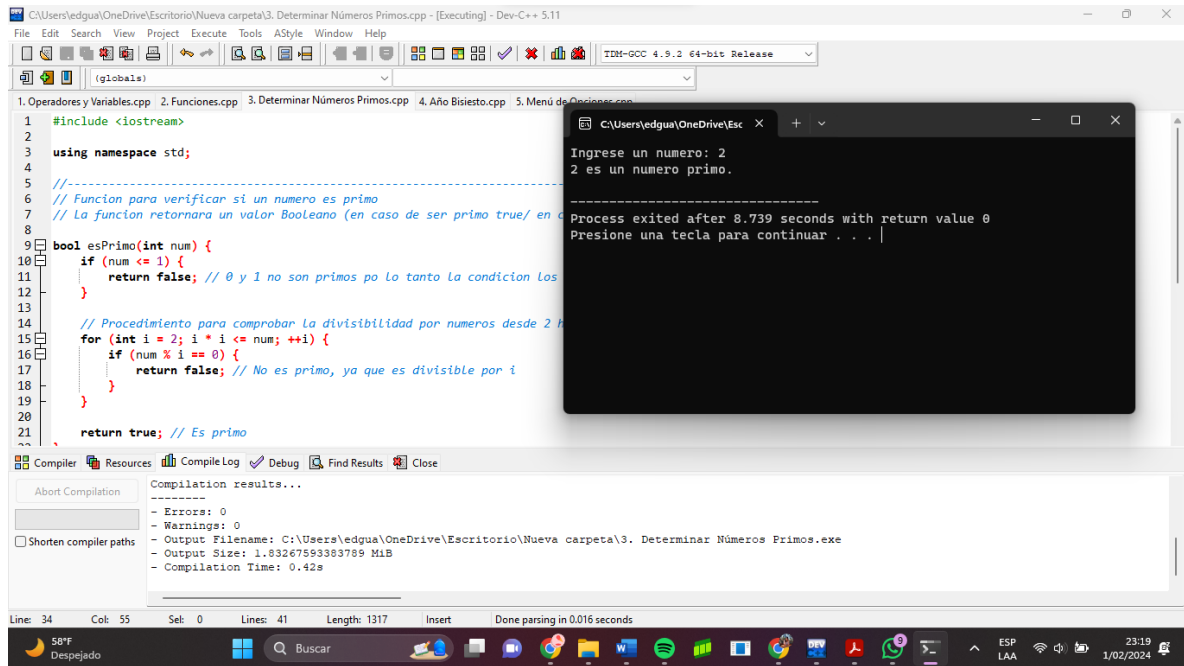
    // Lamar a la funcion esPrimo
    // Mostrando resultado de la funcion esPrimo
    if (esPrimo(num)) {
        cout << num << " es un numero primo." << endl;
    } else {
        cout << num << " no es un numero primo." << endl;
    }

    return 0; // Final del programa
}

```

Valores de entrada sugeridos para una ejecución correcta: números primos aquellos solo tienen 2 factores: 1 y ellos mismos. Por ejemplo, los primeros 5 números son 2, 3, 5, 7 y 11.





4. Año Bisiesto:

El siguiente programa tiene como objetivo verificar si un año es bisiesto.

```
#include <iostream>
```

```
using namespace std;
```

```
//-----  
-----
```

```
// Funcion que verificara si un año es bisiesto
```

```
bool esBisiesto(int year) {
```

```
    // Un año solo es bisiesto si se cumple lo siguiente, es divisible por 4, pero  
    no por 100 a menos que también sea divisible por 400.
```

```
    //condicion con retorno booleano
```

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

```
        return true;
```

```
    } else {
```

```
        return false;
```

```
    }
```

```
}
```

```
//-----  
-----
```

```
int main() {
```

```
    //Declaracion de variables
```

```
    int year;
```

```

        // Solicitud de variables
cout << "Ingrese un año: ";
cin >> year;

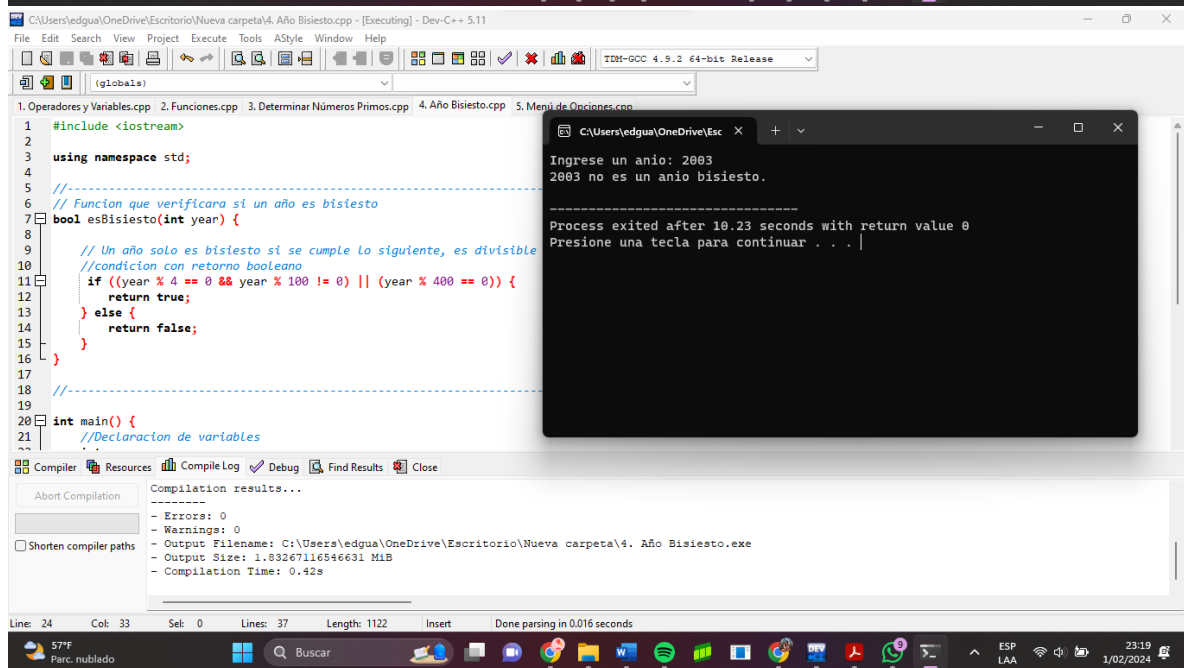
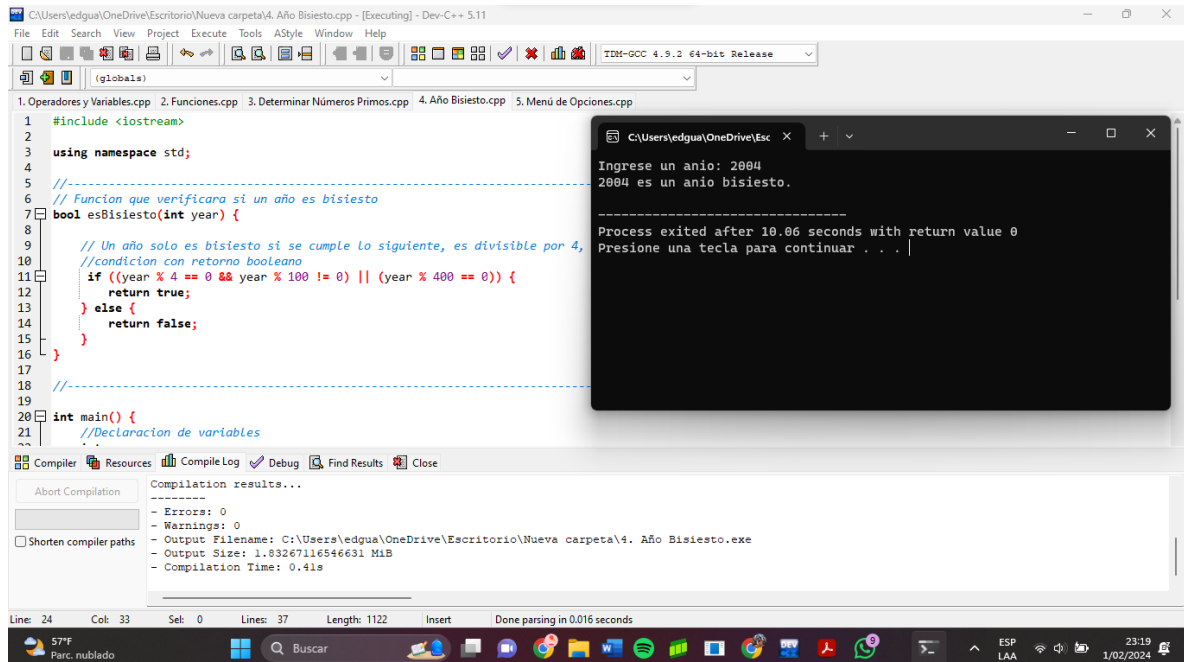
// Llamado a la función esBisiesto
// Mostrando resultado de la función esBisiesto
if (esBisiesto(year)) {
    cout << year << " es un año bisiesto." << endl;
} else {
    cout << year << " no es un año bisiesto." << endl;
}

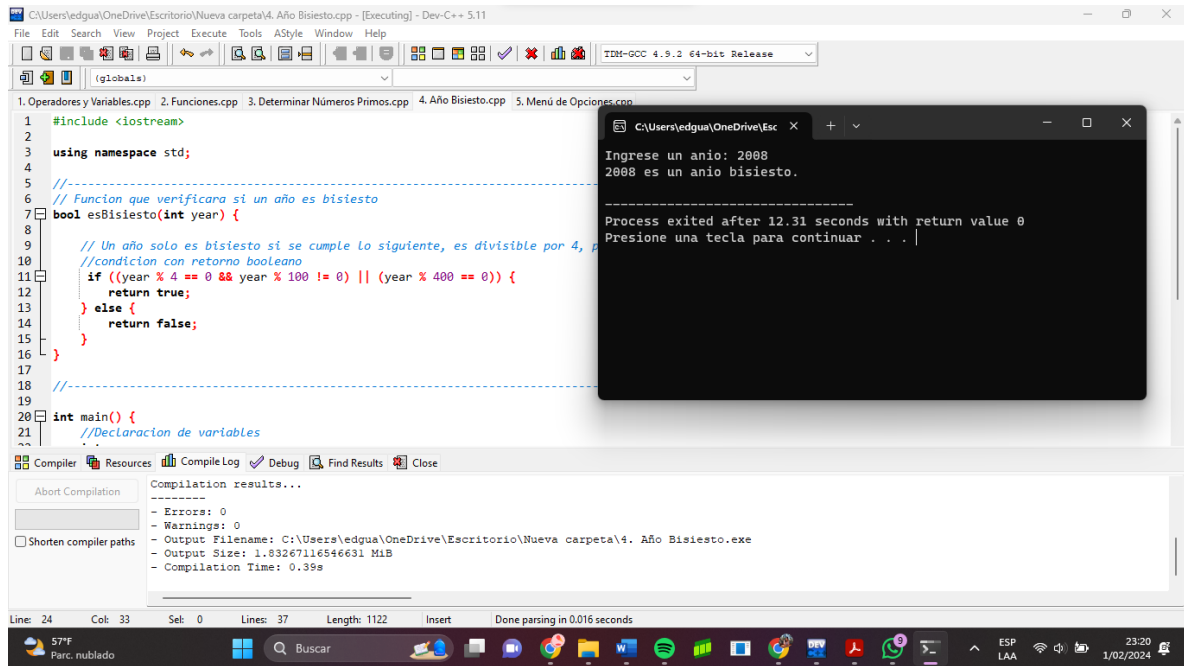
return 0; // Final del programa
}

```

Valores de entrada sugeridos para una ejecución correcta: Años bisiestos. Un año bisiesto es un año solar en el que se produce la intercalación periódica de un día adicional en el propio año, lo que quiere decir:

- Todos los años bisiestos son divisibles entre 4.
- Aquellos años que son divisibles entre 4, pero no entre 100, son bisiestos.
- Los años que son divisibles entre 100, pero no entre 400, no son bisiestos.
- Sin embargo, los años divisibles entre 100 y entre 400 sí que son bisiestos.





5. Menú de Opciones:

El siguiente programa tiene como objetivo un menú que permita al usuario seleccionar entre las diferentes funciones implementadas. Utilizando un bucle para que el usuario pueda realizar varias operaciones sin reiniciar el programa.

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
//-----  
-----
```

```
// Declaraciones de funciones
```

```
void opevar();
```

```
void fun();
```

```
void primo();
```

```
void anio();
```

```
int menu();
```

```
//-----  
-----
```

```
// Funcion para calcular la potencia
```

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

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

```
        return 1.0;
```

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

```
        return 1.0 / pow(base, -exponente);
```

```
    } else {
```

```
        return pow(base, exponente);
```

```
    }
```

```
}
```

```
//*****  
*****
```

```
// Funcion principal main
```

```
int main() {
```

```
    int opcion;
```

```
// Impelmentacion de bucle para evitar salir del programa.
```

```
    do {
```

```
        system("cls");
```

```
        opcion = menu();
```

```
        switch (opcion) {
```

```
            case 1:
```

```
                opevar();
```

```
                break;
```

```
            case 2:
```

```
                fun();
```

```
                break;
```

```
            case 3:
```

```
                primo();
```

```
                break;
```

```
            case 4:
```

```
                anio();
```

```
                break;
```

```
            case 5:
```

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

```

        break;

    default:

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

    }

} while (opcion != 5); // 5 casos

return 0;

}

//*****
*****

// Funcion de menu
int menu() {
    int x;

    cout << "-----MENU DE OPCIONES-----" << endl;
    cout << "Seleccione una de las siguientes opciones:" << endl;
    cout << " 1. Operadores y variables" << endl;
    cout << " 2. Funciones" << endl;
    cout << " 3. Determinacion de numeros primos" << endl;
    cout << " 4. Verificacion de anio bisiestro" << endl;
    cout << " 5. SALIR" << endl;
    cout << endl;

    cout << "Elija una opcion: ";
    cin >> x;

```

return x; //x es la variable que escogeremos y dado el retorno nos enviara directo a la funcion que se desea utilizar

}

```
//*****  
*****
```

// Funciones

void opevar() {

system("cls");

int num1, num2;

cout << "Ingrese el primer numero entero: ";

cin >> num1;

cout << "Ingrese el segundo numero entero: ";

cin >> num2;

int sum = num1 + num2;

int rest = num1 - num2;

int mult = num1 * num2;

float div;

if (num2 != 0) {

div = static_cast<float>(num1) / num2;

} else {

cout << "No se puede dividir entre cero, ERROR." << endl;

return; // No es necesario cerrar el programa después del error, simplemente retorna.

}

```

bool igual = num1 == num2;
bool dif = num1 != num2;
bool mayor = num1 > num2;
bool menor = num1 < num2;


cout << "\nRESULTADOS:" << endl;
cout << "Suma: " << sum << endl;
cout << "Resta: " << rest << endl;
cout << "Multiplicacion: " << mult << endl;
cout << "Division: " << div << endl;


cout << "\nCOMPARACIONES ENTRE VARIABLES:" << std::endl;
cout << num1 << " es igual a " << num2 << "? " << boolalpha << igual <<
endl;
cout << num1 << " es diferente de " << num2 << "? " << boolalpha << dif
<< endl;
cout << num1 << " es mayor que " << num2 << "? " << boolalpha <<
mayor << endl;
cout << num1 << " es menor que " << num2 << "? " << boolalpha <<
menor << endl;


system("pause");
}
//-----
-----

void fun() {
    system("cls");

    float base;

```

```

int exponente;

cout << "Ingrese la base: ";

cin >> base;


cout << "Ingrese el exponente: ";

cin >> exponente;


float resultado = calcularPotencia(base, exponente);

cout << "El resultado de " << base << "^" << exponente << " es: " <<
resultado << endl;


system("pause");
}

//-----
-----

void primo() {
    system("cls");


    int num;

    cout << "Ingrese un numero: ";

    cin >> num;


    bool esPrimo = true;

    if (num <= 1) {
        esPrimo = false;
    } else {
        for (int i = 2; i * i <= num; ++i) {
            if (num % i == 0) {

```



```
        esPrimo = false;
        break;
    }
}
}
```

```
if (esPrimo) {
    cout << num << " es un numero primo." << endl;
} else {
    cout << num << " no es un numero primo." << endl;
}
```

```
    system("pause");
}
```

```
//-----
-----
```

```
void anio() {
    system("cls");
```

```
    int year;
    cout << "Ingrese un anio: ";
    cin >> year;
```

```
    bool esBisiesto = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
```

```
    if (esBisiesto) {
        cout << year << " es un anio bisiesto." << endl;
```

```

} else {

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

}

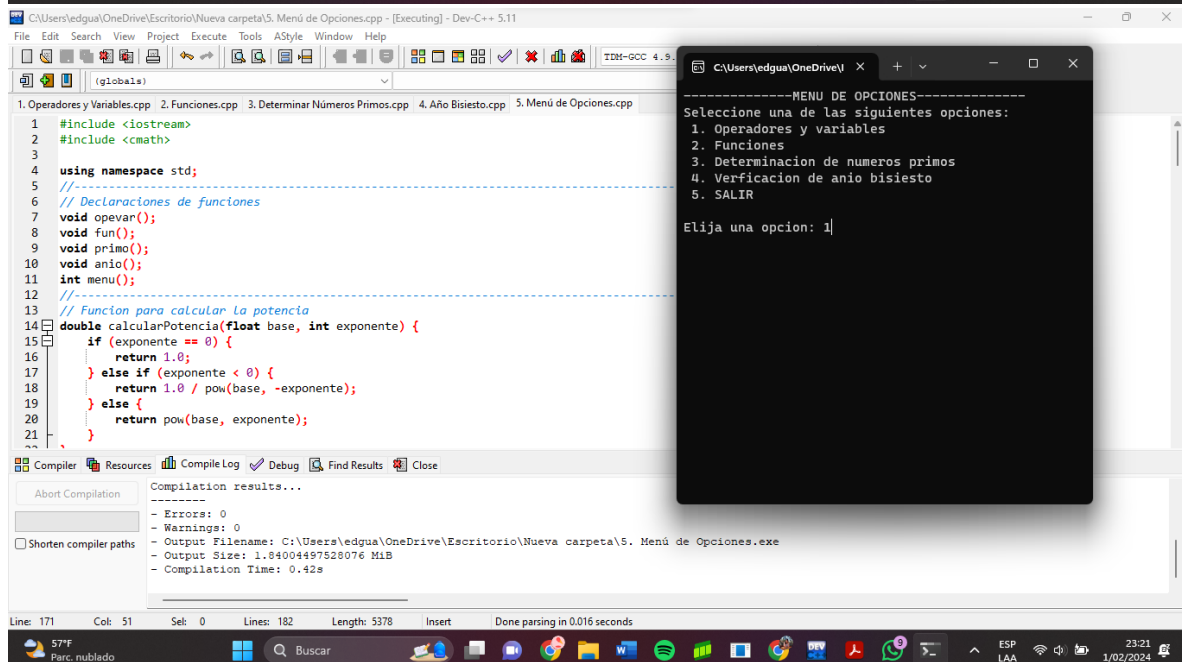
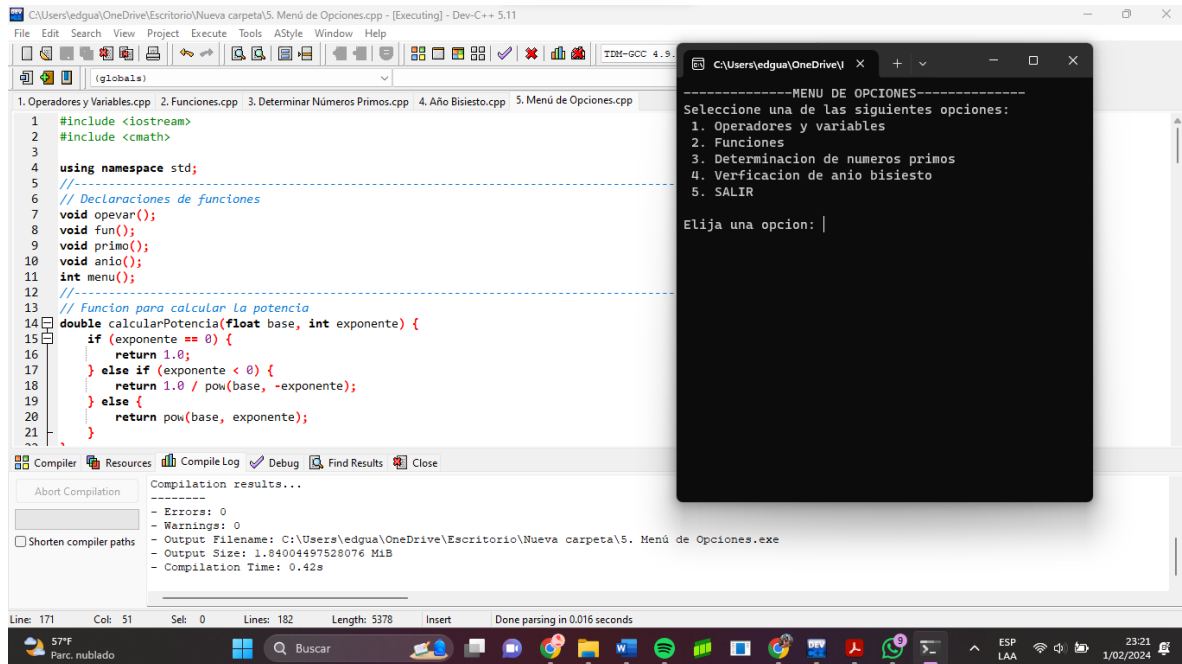
```

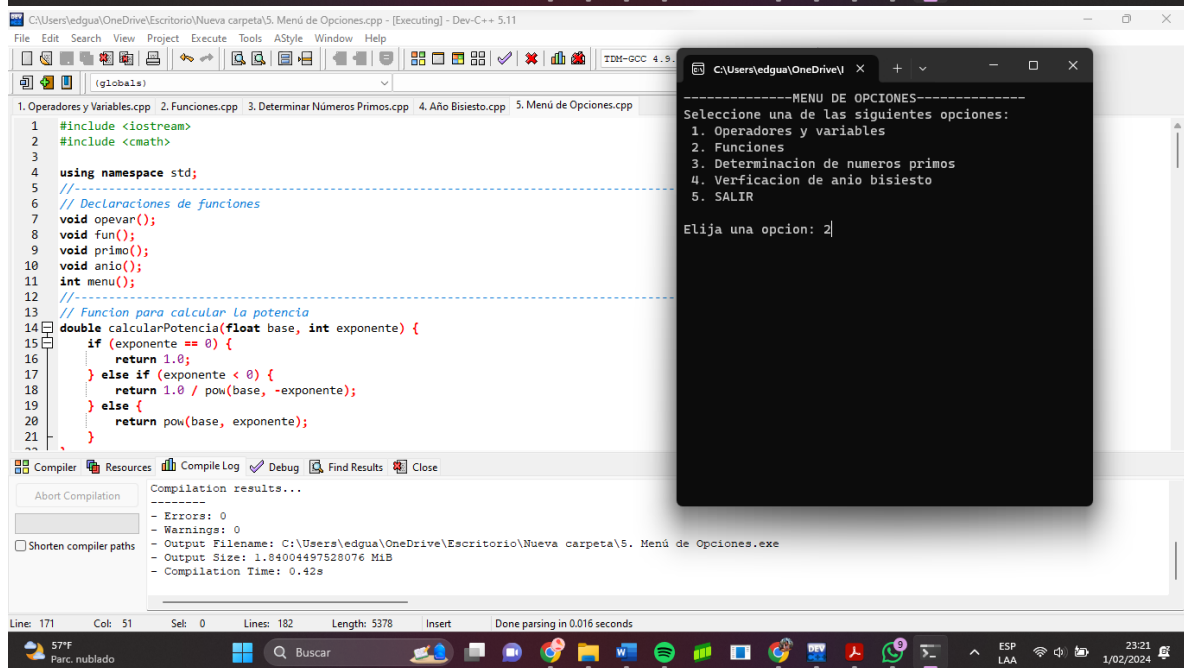
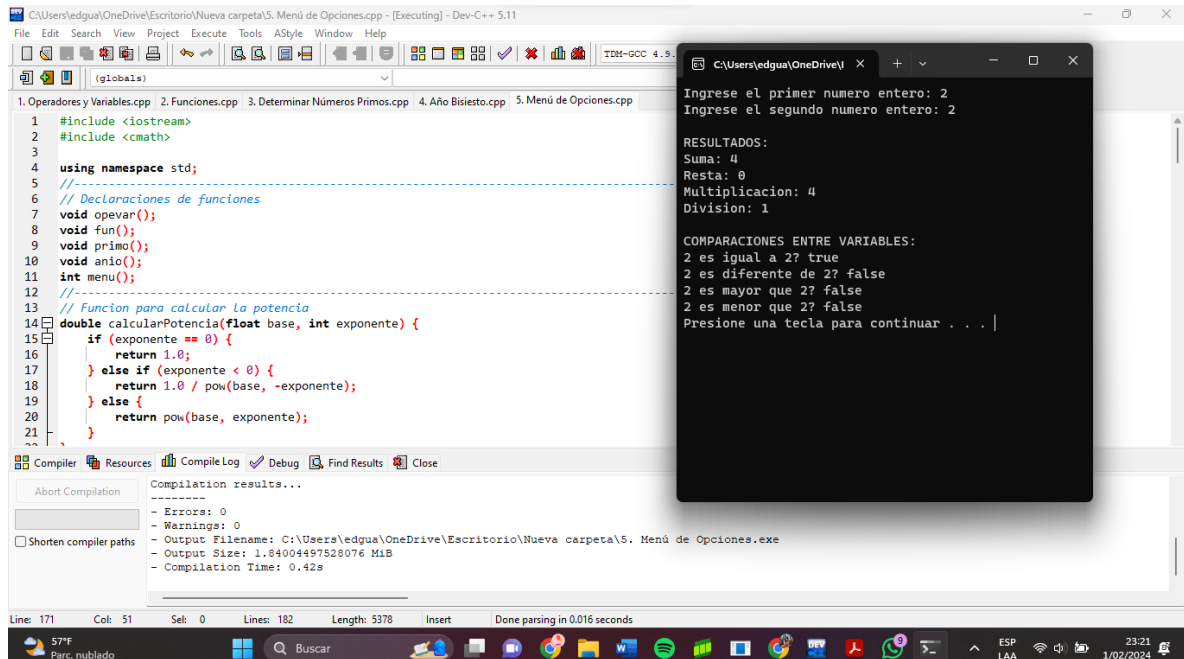
```

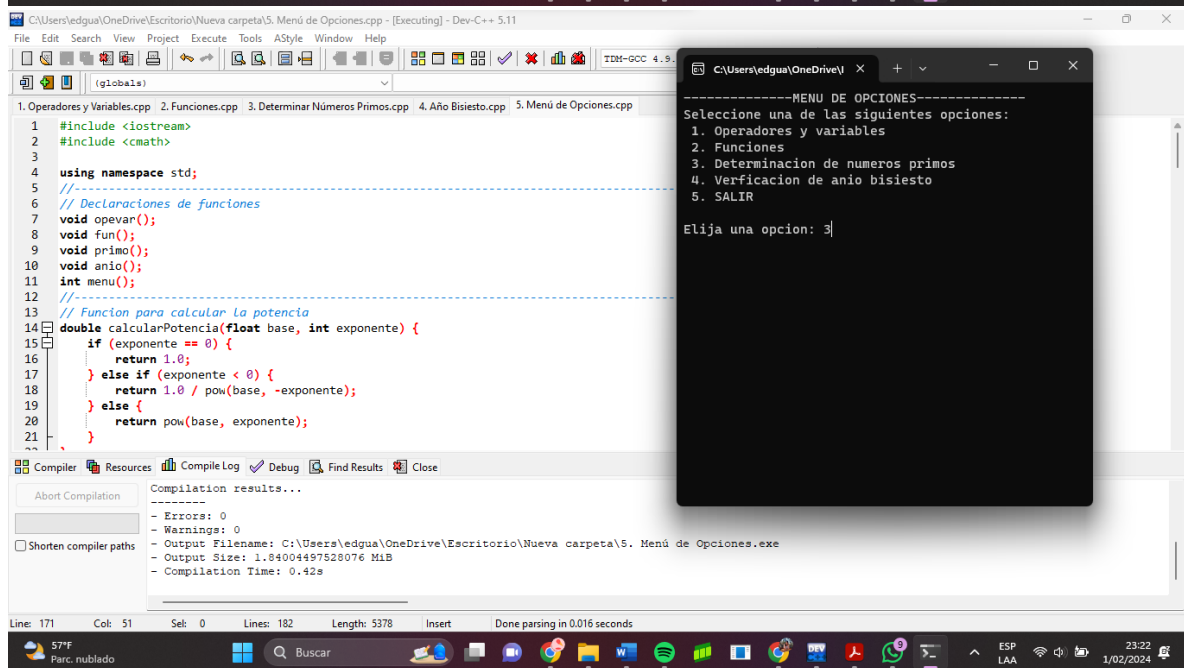
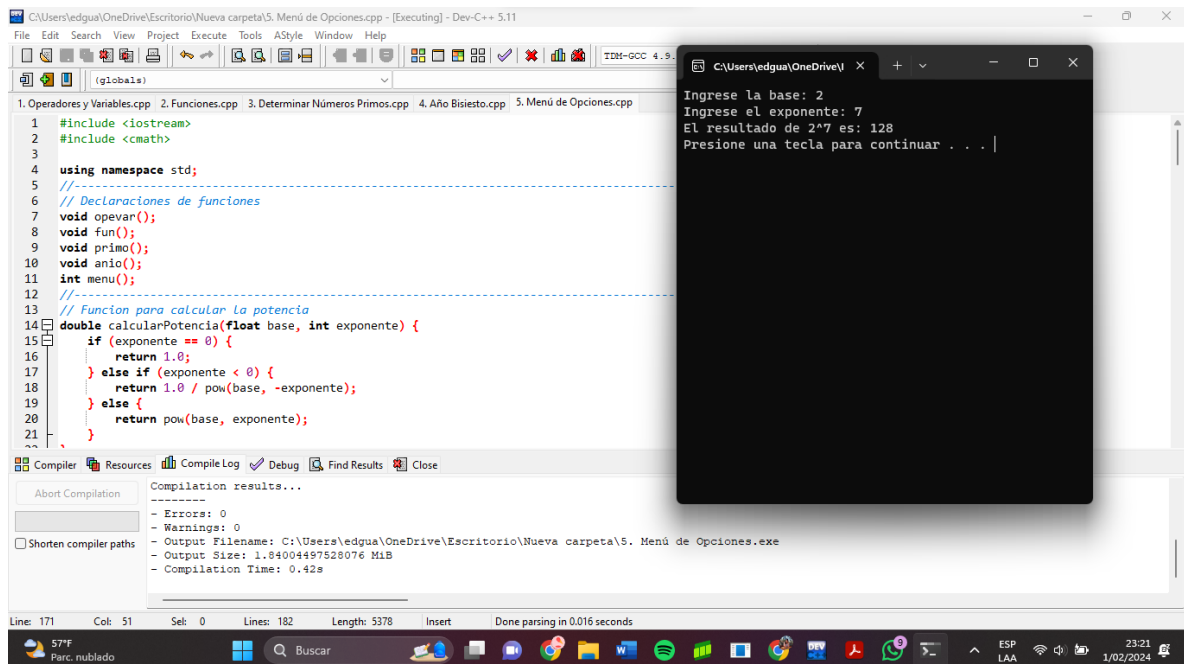
system("pause");

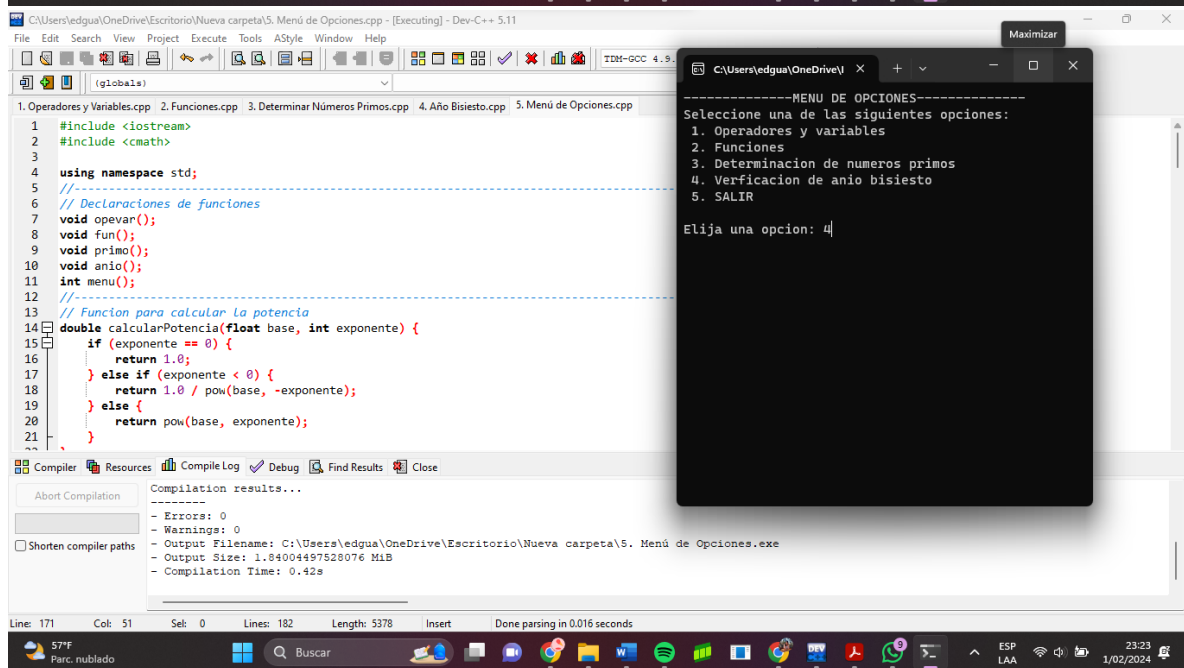
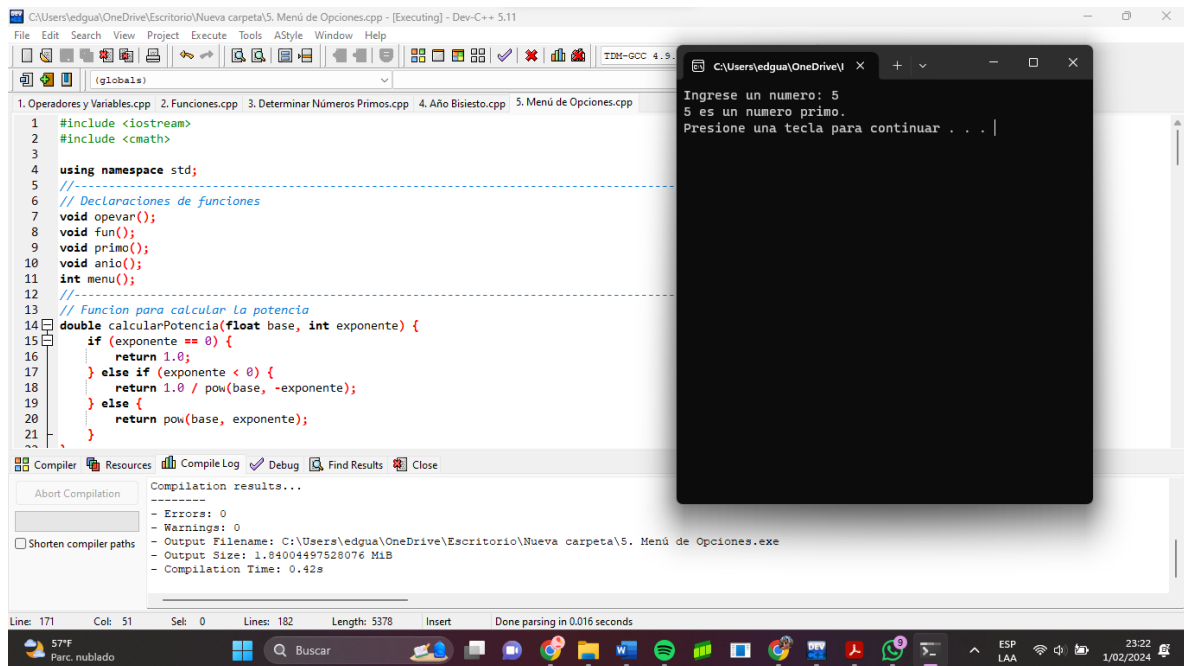
}

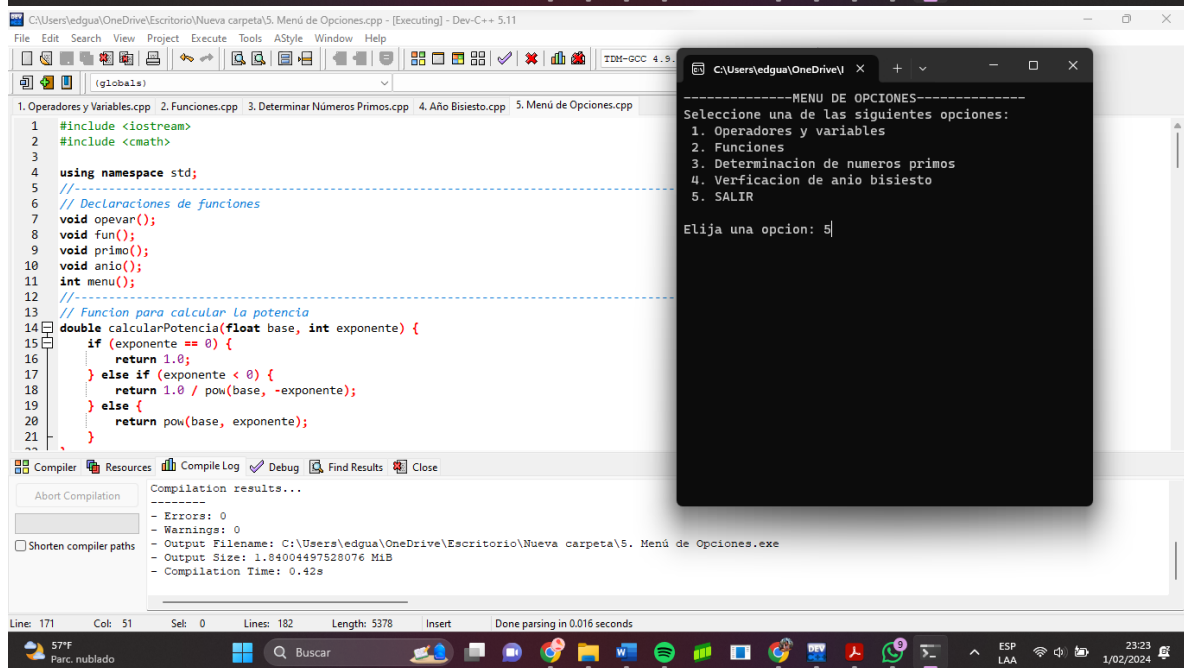
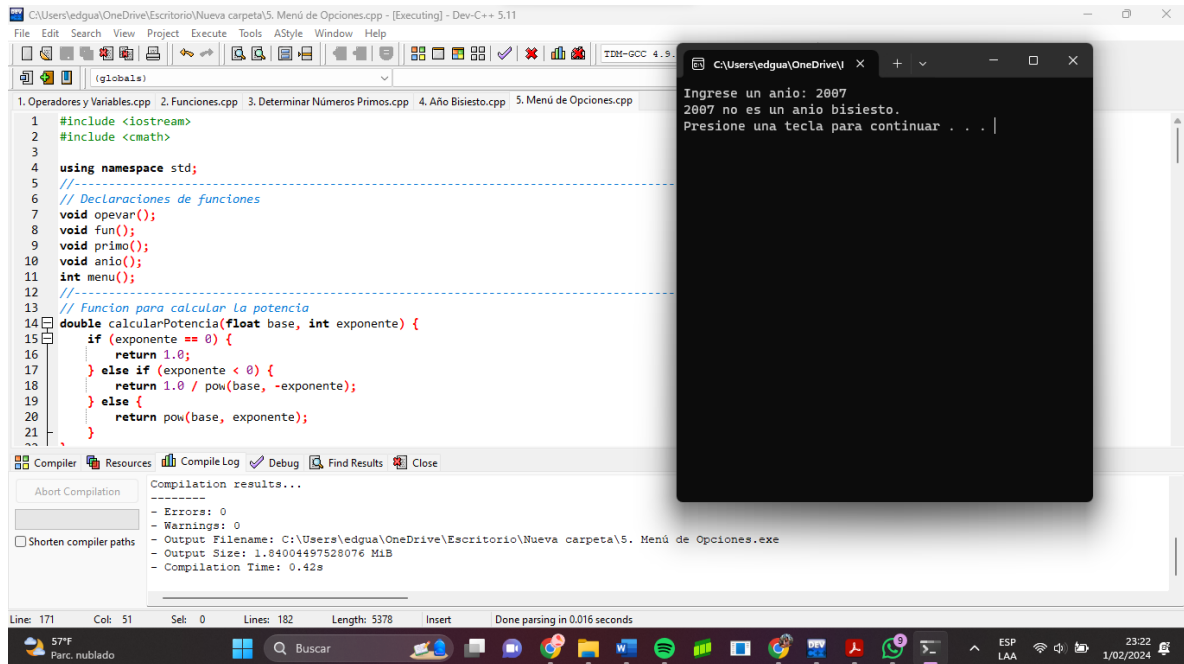
```

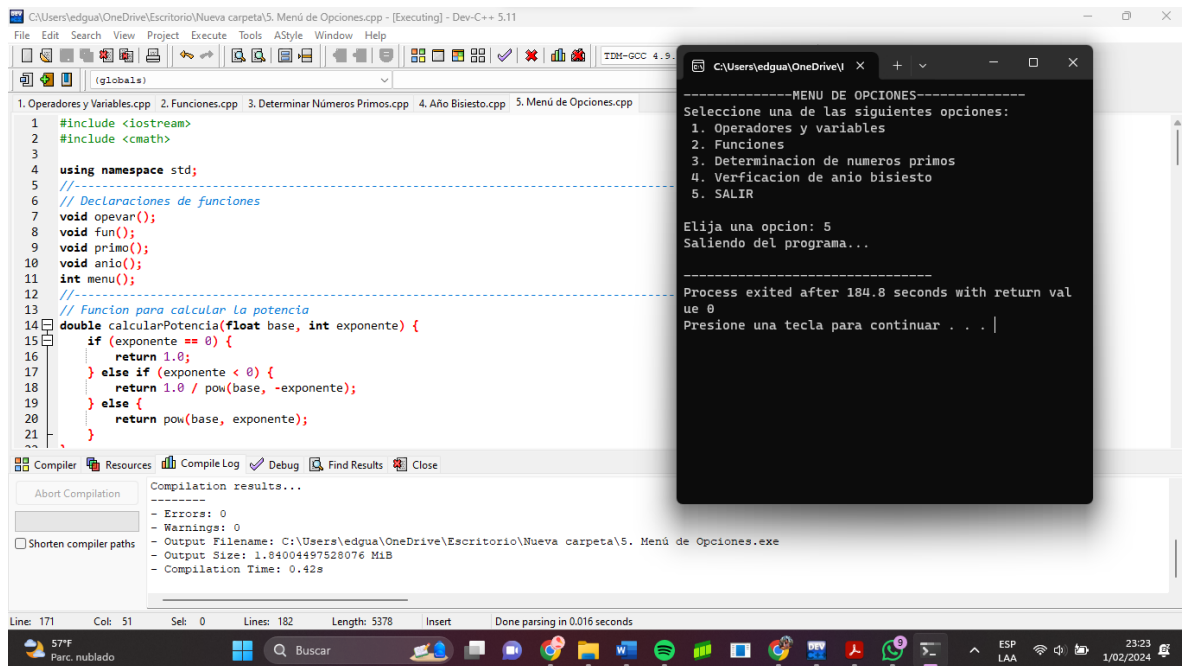












Enlace directo a repositorio de GitHub:

<https://github.com/ousrr/Laboratorios-Programaci-n-I>

Conclusión

En conclusión, este proyecto ha proporcionado una oportunidad para explorar y aplicar conceptos fundamentales de programación en el lenguaje C++. La implementación de un programa que realiza operaciones aritméticas y lógicas básicas, junto con funciones adicionales para determinación la primalidad de un número y verificación si un año es bisiesto, ha permitido consolidar conocimientos en el manejo de variables, operadores y estructuras de control.

La experiencia de desarrollo de este proyecto ha contribuido al fortalecimiento de habilidades. Mejorando de manera significativa la necesidad de manejar situaciones especiales, ha añadido complejidad y profundidad los conocimientos que ya se tenían, proporcionando una valiosa práctica en la resolución de problemas.

Este proyecto ha sido una oportunidad para mejorar la capacidad de comprensión de conceptos técnicos a través de la documentación. La combinación de la teoría y la aplicación práctica enriqueció la comprensión global de los principios de la programación en C++.