



UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

Curso: Programación, Jornada Matutina

Sección: "A"

Catedrática: Ing. Carlos Alejandro Arias.



Carnet: \_\_5090-23-1762

Nombre completo: Roberto Carlos de León Gramajo

Fecha entrega: 22/01/2024

Tema tarea: Trabajo#1

## **Introducción**

Realizar un programa ya sea con Visual Code, o con DevC++, que realice el procedimiento de una calculadora, se debe de ingresar 2 valores, los cuales tienes que pasar por las funciones aritméticas (Suma, Resta, Multiplicación, División).

//Roberto Carlos de León Gramajo 5090-23-1762.

//Primer ejercicio

#include <iostream>

using namespace std;

//Definimos 3 variables, las cuales nos ayudaran en todo el cálculo, dos serán valores que el usuario ingresa, y una tercera es el resultado que estos dos valores tendrán.

float valor1 = 0;

float valor2 = 0;

float resultado = 0;

//Definimos 4 variables independientes para llegar a tener un orden en cada procedimiento, que no tendrán retorno, sino un procedimiento, cada uno tendrá el nombre de una operación aritmética, en estas las definiremos como "void", para saber que serán procedimientos sin retorno.

void suma();

void resta();

void multiplicacion();

void division();

//Nuestro procedimiento central será "int main", es el único que si tendrá un procedimiento con retorno, y nos ayudara a poder tener una respuesta para el usuario, por eso mismo registra los dos valores del usuario que desea agregar, y cada procedimiento sin retorno, tendrá su respuesta independientemente al de los demás.

int main() {

    cout <<"\tCAICULA DE TODO"<< endl;

    cout <<"Ingrese el primer numero"<< endl;

    cin>> valor1;

    cout <<"Ingrese el segundo numero"<< endl;

    cin>> valor2;

    suma();

    return 0;

}

void suma() {

// El primer procedimiento sin retorno que se tendrá será de suma, el cual hacemos una formula aritmética entre los dos valores que ya nos proporciona nuestro "main".

```
cout << "\n\n#Suma" << endl;
resultado = (valor1 + valor2);
cout << "Nuestra suma es: " << resultado << endl;
resta();
}
```

```
void resta() {
    //Mi segundo procedimiento desarrollaremos la formula aritmética de resta.
    cout << "\n\n#Resta" << endl;
    resultado = (valor1 - valor2);
    cout << "Nuestra resta es: " << resultado << endl;
    multiplicacion();
}
```

```
void multiplicacion() {
    // Mi tercer procedimiento desarrollaremos la formula aritmética multiplicación, y a nuestro
    procedimiento sin retorno se llamara multiplicacion.
    cout << "\n\n#Multiplicacion" << endl;
    resultado = (valor1 * valor2);
    cout << "Nuestra multiplicacion: " << resultado << endl;
    division();
}
```

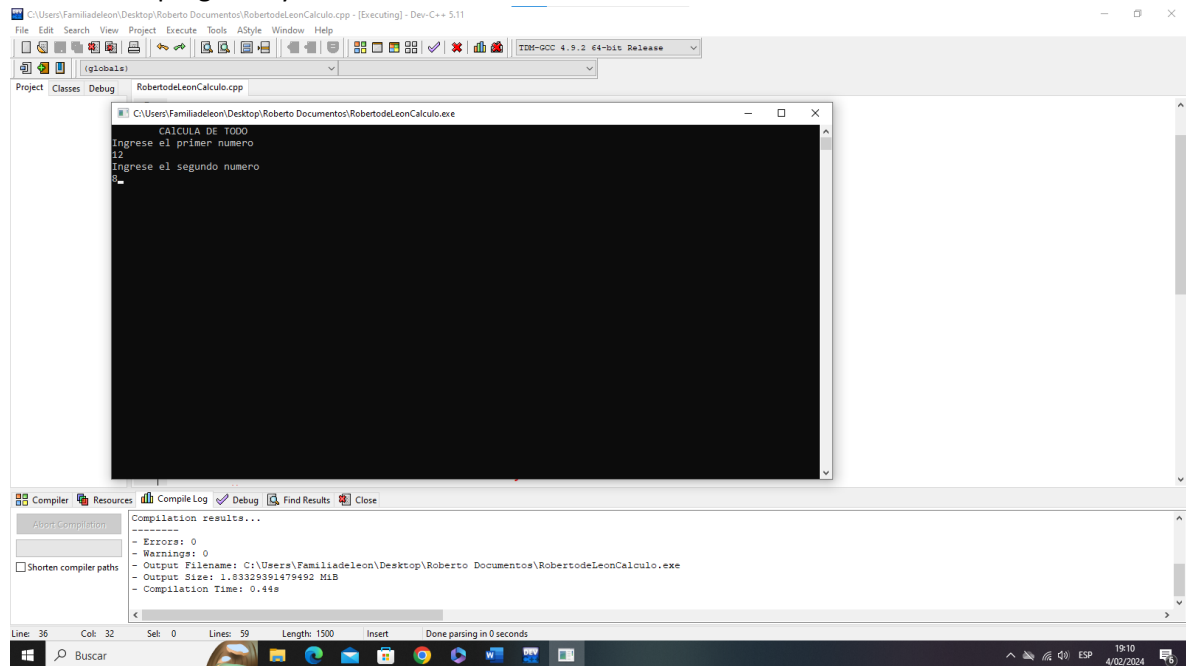
```
void division() {
    //El cuarto procedimiento que desarrollo es sobre la operación aritmética de una división, que
    depende de los valores que el usuario nos brinde, también definimos que el segundo valor si no es
    igual a 0, nos dará un resultado, pero si es igual a cero, nos dará un procedimiento diferente como
    cuando es igual a 0.
    cout << "\n\n#Division" << endl;
```

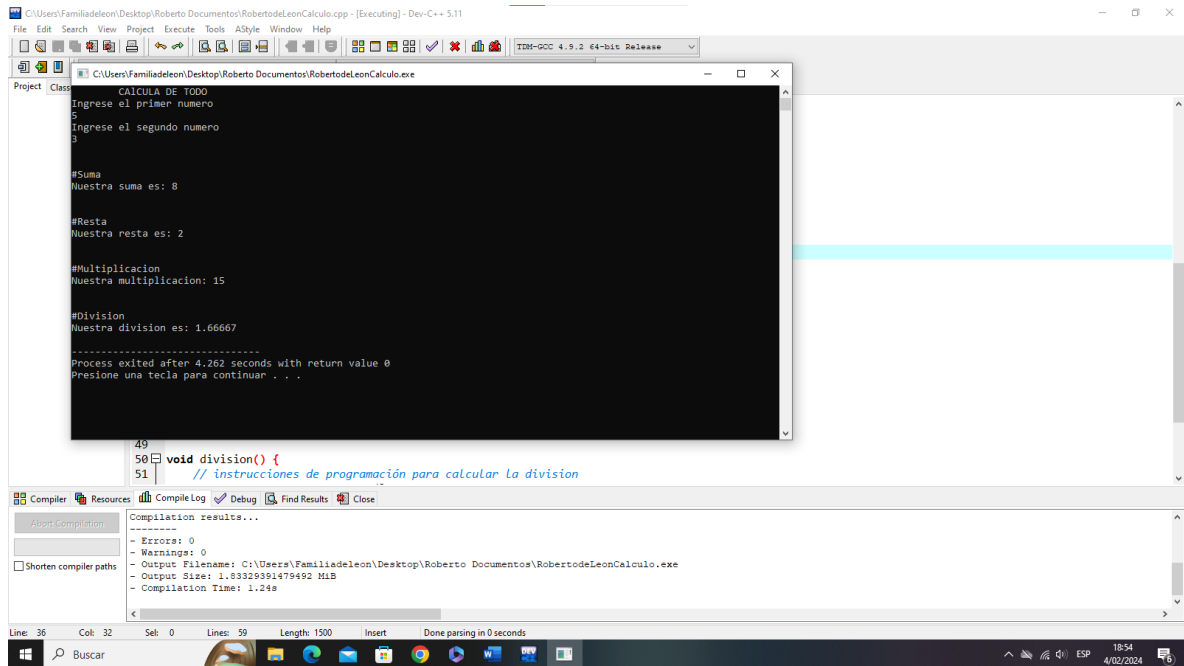
```

if (valor2 != 0) {
    resultado = (valor1 / valor2);
    cout << "Nuestra division es: " << resultado << endl;
} else {
    cout << "No se puede dividir por cero." << endl;
}
}
}

```

### Pruebas del programa ya corriendo en nuestro .exe





## **Conclusión**

Utilice varios procedimientos que no retornen ningún valor con la función (void), las utilice para que cada uno realice un trabajo por separado y tenga cada procedimiento un orden, y así se proyecten en la función principal, realizar suma, resta, multiplicación, y división, las cuales se comunican con la función principal, quien proporciona una variable que será publica para todas las funciones sin retorno.

Las funciones más complicadas fueron la división y la multiplicación, para entender cómo se divide un número con otro, cual debe de ser el divisor, y el dividendo, mientras la multiplicación reconocer números negativos pensé que sería complicado, pero intentando hacer la operación como una la conoce convencionalmente, funciono.