

Universidad Mariano Gálvez de Guatemala

Facultad de ingeniería en sistemas

Campus Villa nueva, Guatemala

Curso: Programación I

Docente: Inge. Carlos Arias



Actividad: Laboratorio 5

Nombre: Stephanie Cristina Sabán Cárcamo

Carnet: 5090- 23-11167

Sección: "A"

Fecha: 19/02/24

Introducción

Este informe tiene como objetivo hacer una recopilación de la compilación, ejecución y pruebas del código del programa que conforma el laboratorio 5, de la misma forma también se incluye el link del repositorio donde está subido el código en extensión .cpp y .exe, también en las capturas del código está explicado por comentarios como está constituido el programa.

Este laboratorio se trata de un programa que tenga un menú de opciones.

1. Suma de números pares e impares:

Escribe una función que tome un número entero como parámetro y devuelva la suma de todos los números pares o impares (dependiendo de una condición dada) hasta ese número utilizando un bucle for.

2. Factorial con bucle while:

Implementa una función que calcule el factorial de un número dado utilizando un bucle while.

El factorial de un número entero positivo n , denotado como $n!$, es el producto de todos los enteros positivos menores o iguales a n . Por ejemplo, el factorial de 5 ($5!$) se calcula como $5 \times 4 \times 3 \times 2 \times 1$, que es igual a 120.

3. Contador de dígitos:

Crea una función que reciba un número entero y cuente la cantidad de dígitos que tiene. Utiliza un bucle do-while para garantizar que el número se procese al menos una vez.

4. Ordenamiento de números (ascendente):

Diseña una función que ordene tres números enteros de menor a mayor utilizando condicionales y métodos de intercambio, sin usar arrays. Puedes utilizar bucles for, while, o do-while según prefiera.

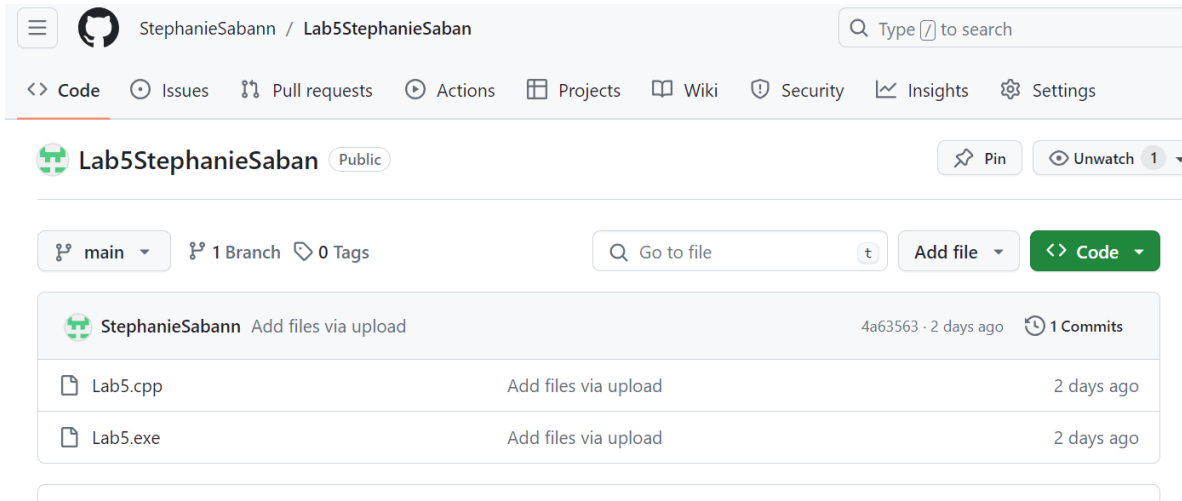
5. Suma de dígitos de un número:

Implementa una función que sume todos los dígitos de un número entero dado. Utiliza bucles for, while, o do-while según prefieras para iterar sobre los dígitos del número.

Dirección del repositorio (link)

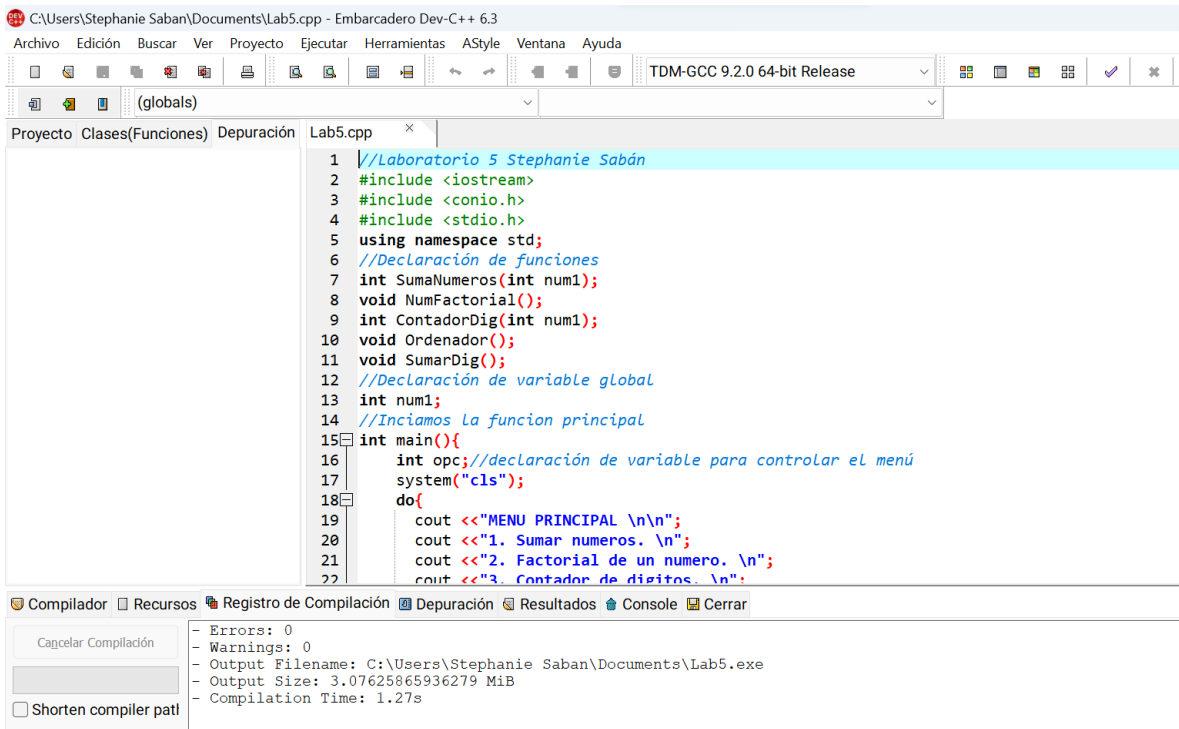
El laboratorio 1 está almacenado en el siguiente repositorio, el archivo tiene nombre de “Lab5”.

<https://github.com/StephanieSabann/Lab5StephanieSaban.git>



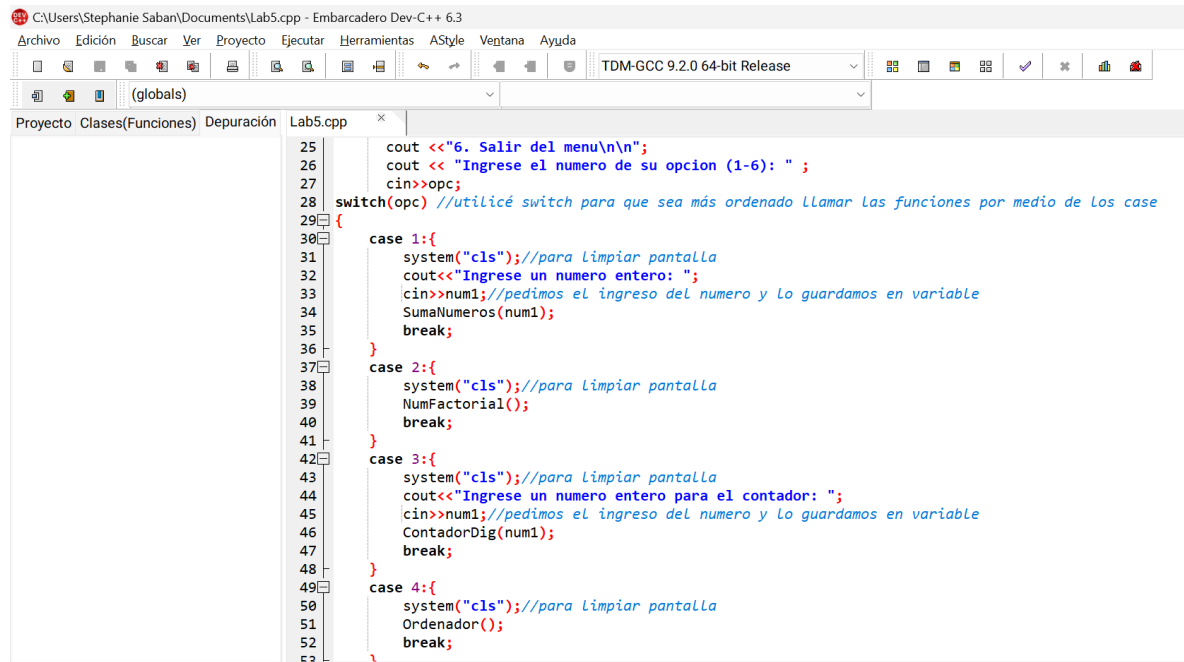
Pruebas de compilación y Código comentado

En esta prueba de compilación podemos ver que no hay errores, ni advertencias.



Empezamos colocando las bibliotecas correspondientes Declaración de variables, Se declaran las variables fact, factorial, a, num1, num2, num3, opc, suma, y ParIm para almacenar valores y controlar el flujo del programa, se definen las funciones SumaNumeros, NumFactorial, ContadorDig, Ordenador, y SumarDig que realizan operaciones específicas.

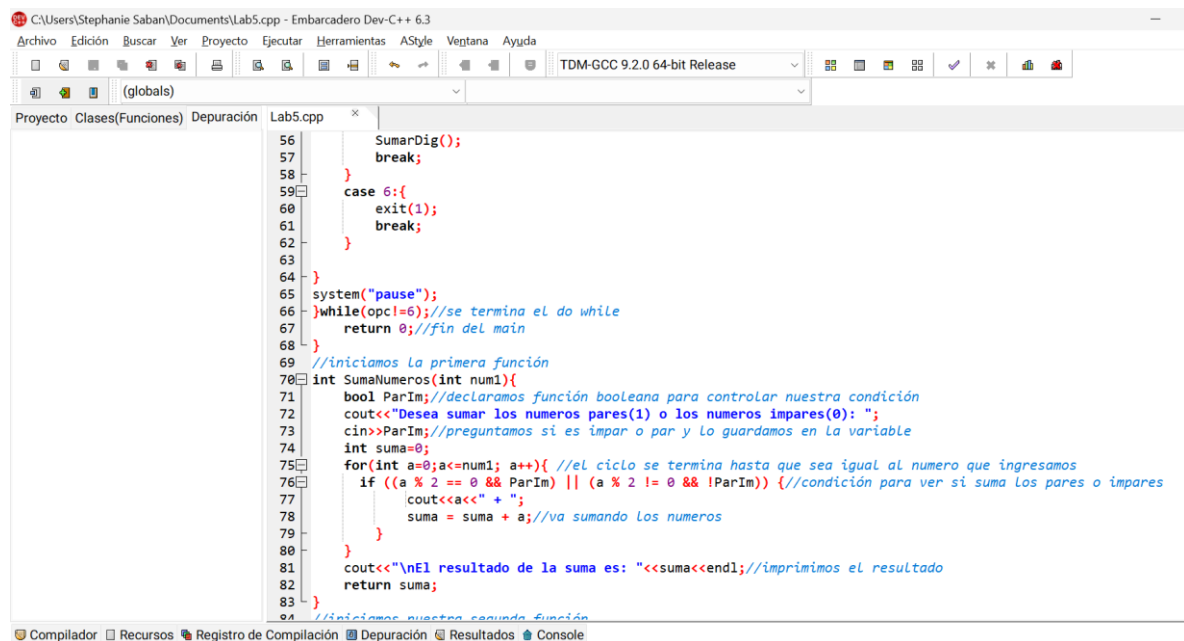
En el menú principal se muestra un menú con diferentes opciones para realizar operaciones, el usuario selecciona una opción y se ejecuta la operación correspondiente.



```

25      cout << "6. Salir del menu\n\n";
26      cout << "Ingrese el numero de su opcion (1-6): ";
27      cin >> opc;
28      switch(opc) //utilicé switch para que sea más ordenado llamar las funciones por medio de los case
29      {
30      case 1:{
31          system("cls");//para limpiar pantalla
32          cout<<"Ingrese un numero entero: ";
33          cin>>num1;//pedimos el ingreso del numero y lo guardamos en variable
34          SumaNumeros(num1);
35          break;
36      }
37      case 2:{
38          system("cls");//para limpiar pantalla
39          NumFactorial();
40          break;
41      }
42      case 3:{
43          system("cls");//para limpiar pantalla
44          cout<<"Ingrese un numero entero para el contador: ";
45          cin>>num1;//pedimos el ingreso del numero y lo guardamos en variable
46          ContadorDig(num1);
47          break;
48      }
49      case 4:{
50          system("cls");//para limpiar pantalla
51          Ordenador();
52          break;
53      }

```

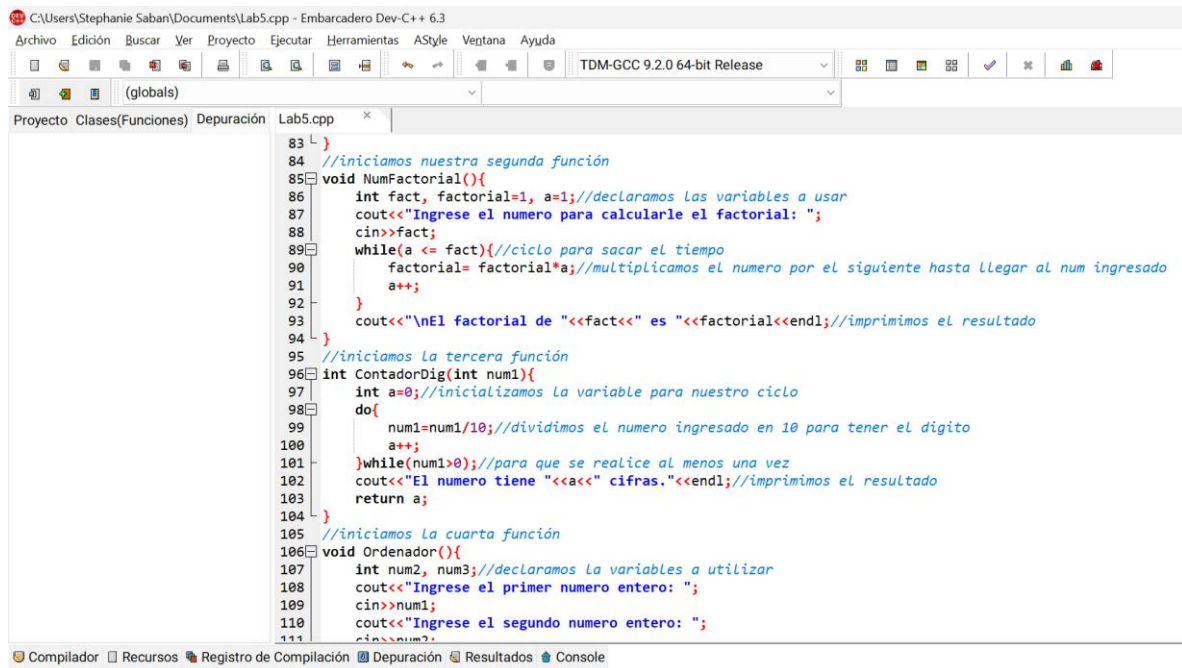


```

56      SumarDig();
57      break;
58      }
59      case 6:{
60          exit(1);
61          break;
62      }
63      }
64      system("pause");
65      while(opc!=6);//se termina el do while
66      return 0;//fin del main
67      }
68      //iniciamos la primera función
69      int SumaNumeros(int num1){
70      bool ParIm;//declaramos función booleana para controlar nuestra condición
71      cout<<"Desea sumar los numeros pares(1) o los numeros impares(0): ";
72      cin>>ParIm;//preguntamos si es impar o par y lo guardamos en la variable
73      int suma=0;
74      for(int a=0;a<=num1;a++){ //el ciclo se termina hasta que sea igual al numero que ingresamos
75      if ((a % 2 == 0 && ParIm) || (a % 2 != 0 && !ParIm)) //condición para ver si suma los pares o impares
76      {
77          cout<<a<<" ";
78          suma = suma + a;//va sumando los numeros
79      }
80      }
81      cout<<"\nEl resultado de la suma es: "<<suma<<endl;//imprimimos el resultado
82      return suma;
83      }
84      //iniciamos nuestra segunda función

```

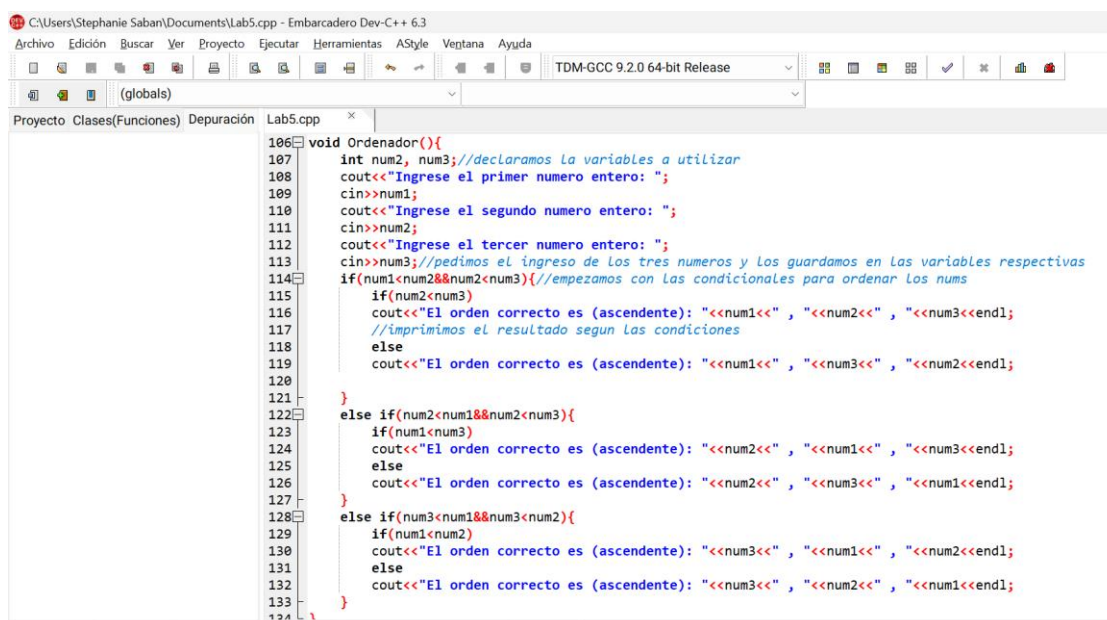
En la primera función se solicita al usuario ingresar un número entero y se pregunta si se desean sumar los números pares o impares. Se realiza la suma de los números pares o impares hasta el número ingresado.



```
83 }
84 //iniciamos nuestra segunda función
85 void NumFactorial(){
86     int fact, factorial=1, a=1;//declaramos Las variables a usar
87     cout<<"Ingrese el numero para calcularle el factorial: ";
88     cin>>fact;
89     while(a <= fact){//ciclo para sacar el tiempo
90         factorial= factorial*a;//multiplicamos el numero por el siguiente hasta llegar al num ingresado
91         a++;
92     }
93     cout<<"\nEl factorial de "<<fact<<" es "<<factorial<<endl;//imprimimos el resultado
94 }
95 //iniciamos la tercera función
96 int ContadorDig(int num1){
97     int a=0;//inicializamos la variable para nuestro ciclo
98     do{
99         num1=num1/10;//dividimos el numero ingresado en 10 para tener el digito
100         a++;
101     }while(num1>0);//para que se realice al menos una vez
102     cout<<"El numero tiene "<<a<<" cifras."<<endl;//imprimimos el resultado
103     return a;
104 }
105 //iniciamos la cuarta función
106 void Ordenador(){
107     int num2, num3;//declaramos la variables a utilizar
108     cout<<"Ingrese el primer numero entero: ";
109     cin>>num1;
110     cout<<"Ingrese el segundo numero entero: ";
111     cin>>num2;
```

Para la segunda función que es factorial de un número: Se solicita al usuario ingresar un número entero para calcular su factorial. Se calcula el factorial del número ingresado por medio de un while que evalúa la condición.

Para la tercera función que es un contador de dígitos: Se pide al usuario ingresar un número entero para contar sus dígitos, se cuenta la cantidad de dígitos en el número ingresado por medio de un DO WHILE.



```
106 void Ordenador(){
107     int num2, num3;//declaramos la variables a utilizar
108     cout<<"Ingrese el primer numero entero: ";
109     cin>>num1;
110     cout<<"Ingrese el segundo numero entero: ";
111     cin>>num2;
112     cout<<"Ingrese el tercer numero entero: ";
113     cin>>num3;//pedimos el ingreso de los tres numeros y los guardamos en las variables respectivas
114     if(num1<num2&&num2<num3){//empezamos con las condicionales para ordenar los nums
115         if(num2<num3)
116             cout<<"El orden correcto es (ascendente): "<<num1<<" , "<<num2<<" , "<<num3<<endl;
117         //imprimimos el resultado segun las condiciones
118         else
119             cout<<"El orden correcto es (ascendente): "<<num1<<" , "<<num3<<" , "<<num2<<endl;
120     }
121 }
122 else if(num2<num1&&num2<num3){
123     if(num1<num3)
124         cout<<"El orden correcto es (ascendente): "<<num2<<" , "<<num1<<" , "<<num3<<endl;
125     else
126         cout<<"El orden correcto es (ascendente): "<<num2<<" , "<<num3<<" , "<<num1<<endl;
127 }
128 else if(num3<num1&&num3<num2){
129     if(num1<num2)
130         cout<<"El orden correcto es (ascendente): "<<num3<<" , "<<num1<<" , "<<num2<<endl;
131     else
132         cout<<"El orden correcto es (ascendente): "<<num3<<" , "<<num2<<" , "<<num1<<endl;
133 }
134 }
```

En la cuarta función se hace uso de los condicionales IF, se solicita al usuario ingresar tres números enteros para ordenarlos de forma ascendente, se ordenan los números ingresados y se muestra el resultado.

```

118         else
119             cout<<"El orden correcto es (ascendente): "<<num1<<" , "<<num3<<" , "<<num2<<endl;
120
121     }
122     else if(num2<num1&&num2<num3){
123         if(num1<num3)
124             cout<<"El orden correcto es (ascendente): "<<num2<<" , "<<num1<<" , "<<num3<<endl;
125         else
126             cout<<"El orden correcto es (ascendente): "<<num2<<" , "<<num3<<" , "<<num1<<endl;
127     }
128     else if(num3<num1&&num3<num2){
129         if(num1<num2)
130             cout<<"El orden correcto es (ascendente): "<<num3<<" , "<<num1<<" , "<<num2<<endl;
131         else
132             cout<<"El orden correcto es (ascendente): "<<num3<<" , "<<num2<<" , "<<num1<<endl;
133     }
134 }
135 //iniciamos La 5ta función
136 void SumarDig(){
137     cout<<"Ingrese un numero entero: ";
138     cin>>num1;//pedimos el ingreso del numero y lo guardamos en la variable
139     int suma=0;//inicializamos la variable para el ciclo
140     while(num1>0){
141         suma= suma + num1%10;//hacemos la suma de los digitos por medio de separar el ultimo digito
142         num1=num1/10;//dividimos entre 10 para quitar el ultimo digito.
143     }
144     cout<<"La suma de los digitos es: "<<suma<<endl;//imprimir el resultado
145 }
146 }

```

Y Para la ultima función utilizamos un while Se pide al usuario ingresar un número entero para sumar sus dígitos, se suman los dígitos del número ingresado y se muestra el resultado.

Pruebas de funcionamiento

```

MENU PRINCIPAL
1. Sumar numeros.
2. Factorial de un numero.
3. Contador de digitos.
4. Ordenamiento de numeros.
5. Sumar los digitos de un numero.
6. Salir del menu

Ingrese el numero de su opcion (1-6): 1

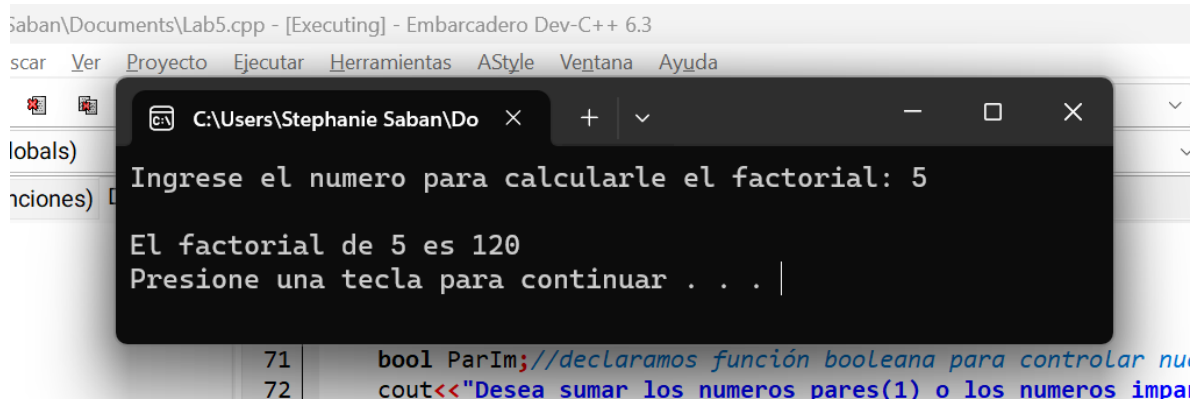
```

```

Ingrese un numero entero: 12
Desea sumar los numeros pares(1) o los numeros impares(0):
1
0 + 2 + 4 + 6 + 8 + 10 + 12 +
El resultado de la suma es: 42
Presione una tecla para continuar . . . |

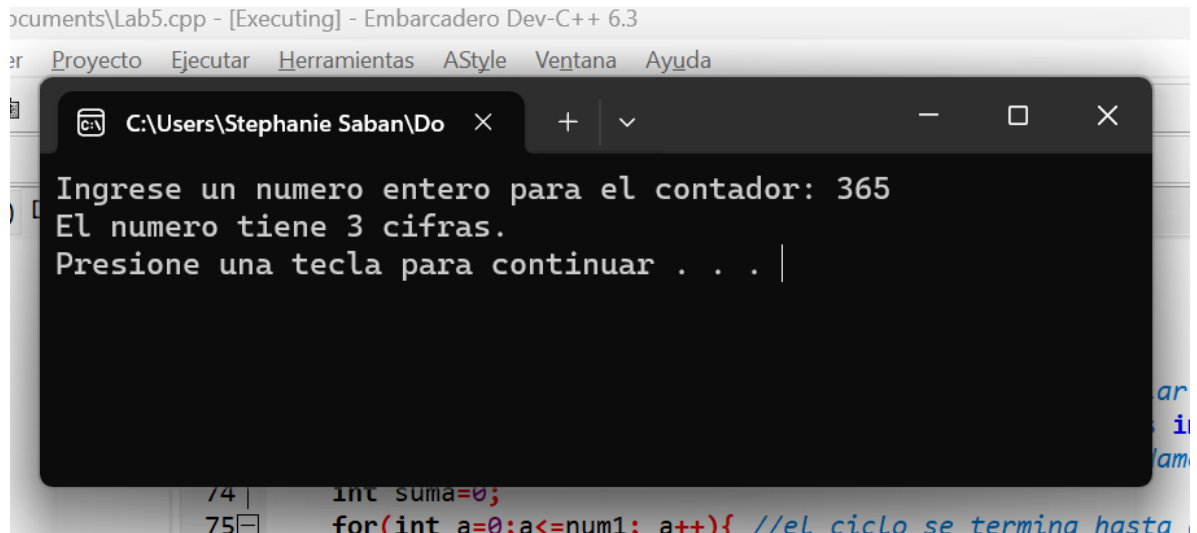
```

Opción 2



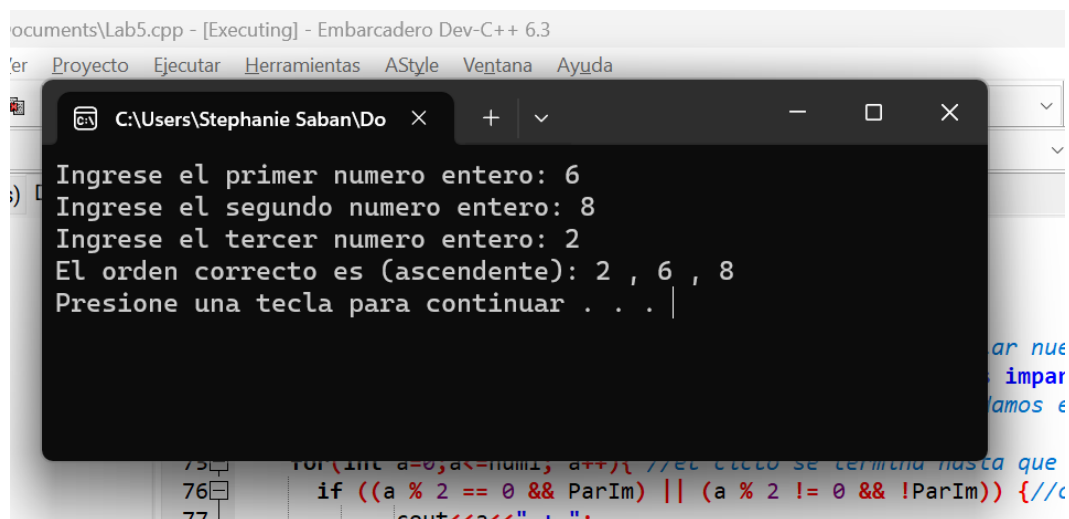
```
laban\Documents\Lab5.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Project Ejecutar Herramientas AStyle Ventana Ayuda
C:\Users\Stephanie Saban\Do x + v - □ x
Ingrese el numero para calcularle el factorial: 5
El factorial de 5 es 120
Presione una tecla para continuar . . . |
71 bool ParIm; //declaramos función booleana para controlar nu
72 cout<<"Desea sumar los numeros pares(1) o los numeros impar
```

Opción 3



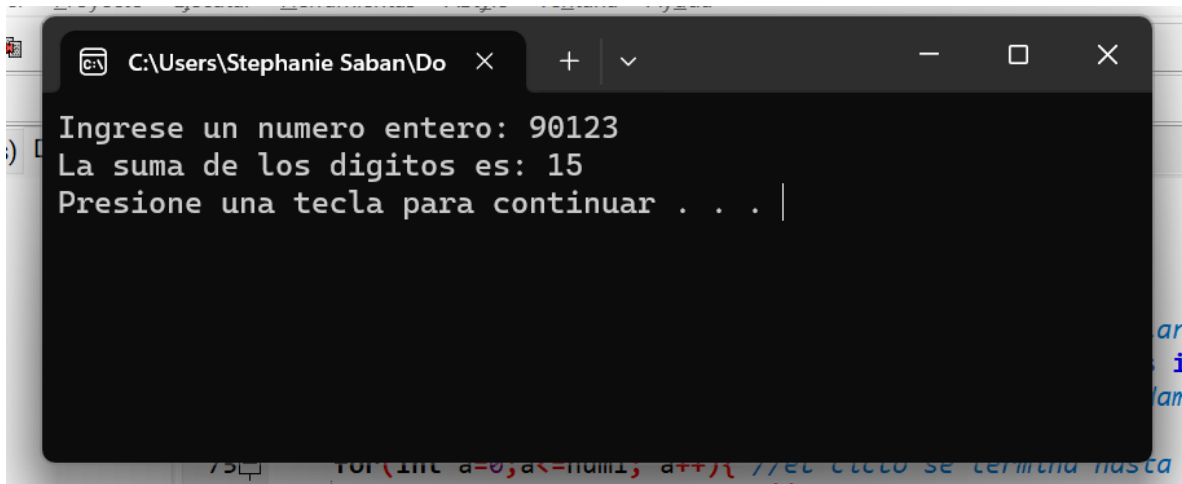
```
Documents\Lab5.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Project Ejecutar Herramientas AStyle Ventana Ayuda
C:\Users\Stephanie Saban\Do x + v - □ x
Ingrese un numero entero para el contador: 365
El numero tiene 3 cifras.
Presione una tecla para continuar . . . |
74 int suma=0;
75 for(int a=0;a<=num1; a++){ //el ciclo se termina hasta
```

Opción 4



```
Documents\Lab5.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Project Ejecutar Herramientas AStyle Ventana Ayuda
C:\Users\Stephanie Saban\Do x + v - □ x
Ingrese el primer numero entero: 6
Ingrese el segundo numero entero: 8
Ingrese el tercer numero entero: 2
El orden correcto es (ascendente): 2 , 6 , 8
Presione una tecla para continuar . . . |
75 for(int a=0;a<=num1; a++){ //el ciclo se termina hasta que
76 if ((a % 2 == 0 && ParIm) || (a % 2 != 0 && !ParIm)) {//c
77 cout<<a<<" + "
```

Opción 5



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Users\Stephanie Saban\Do' and standard window controls. The command prompt displays the following text:

```
Ingresa un numero entero: 90123
La suma de los digitos es: 15
Presione una tecla para continuar . . . |
```

The text is displayed in a monospaced font on a black background. The cursor is positioned at the end of the third line, after the vertical bar.

Conclusión

Este laboratorio es importante ya que demuestra la versatilidad y utilidad de utilizar ciclos y condicionales en la programación para realizar diferentes operaciones matemáticas de manera eficiente. Los ciclos permiten repetir tareas de forma automática, como sumar números pares o impares, contar dígitos o sumar los dígitos de un número. Por otro lado, los condicionales son fundamentales para controlar el flujo del programa y ejecutar diferentes acciones según las decisiones del usuario, como calcular el factorial de un número o ordenar números. En conjunto, el uso de ciclos y condicionales que se usa en este laboratorio facilita la implementación de diversas funcionalidades matemáticas de manera estructurada y organizada.