# https://yt3.ggpht.com/-hYkspDLYXMQ/AAAAAAAAAAI/AAAAAAAAAAA/t8z0vauo_7A/s900-c-k-no/photo.jpghttp://intranet.epn.edu.ec/portal/images/articulos/buo_epn.jpgESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**PROGRAMACION TDSD214**

|  |  |
| --- | --- |
| ASIGNATURA: | Programación |
| PROFESOR: | Ing. Ivonne Maldonado |
| PERÍODO ACADÉMICO: | 2023-B |

# LABORATORIO - 6

|  |
| --- |
| TÍTULO:  **ESTRUCTURAS ITERATIVAS** |
|  |
| Eliminando ciclos que se repetem na vida.. |

**Nombre: Guerra Lovato Josué Eduard**

Tecnología Superior en Desarrollo de Software Laboratorio 6 – Estructuras Iterativas

## PROPÓSITO DE LA PRÁCTICA

Familiarizar al estudiante con las estructuras iterativas (repetitivas) en el lenguaje C++.

## OBJETIVO GENERAL

Conocer las estructuras iterativas mediante la práctica y desarrollo de ejercicios aplicándolo en problemas algo cotidianos el cual servirá para entender de forma más clara cada estructura.

## OBJETIVOS ESPECÍFICOS

* Trabajar con while.
* Trabajar con do-while.
* Trabajar con for.

## INSTRUCCIONES

Realice cada uno de los ejercicios solicitados por su profesor. Utilice:

## OPERADORES

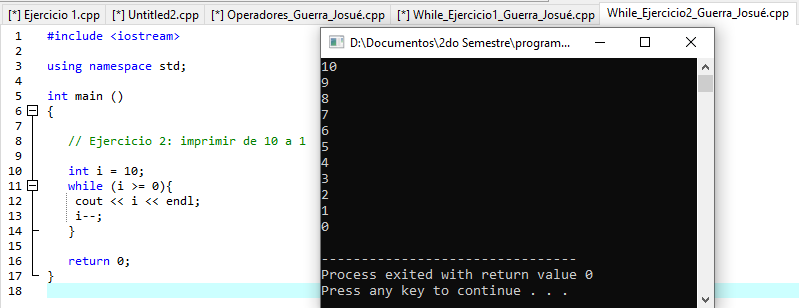
## 

## WHILE

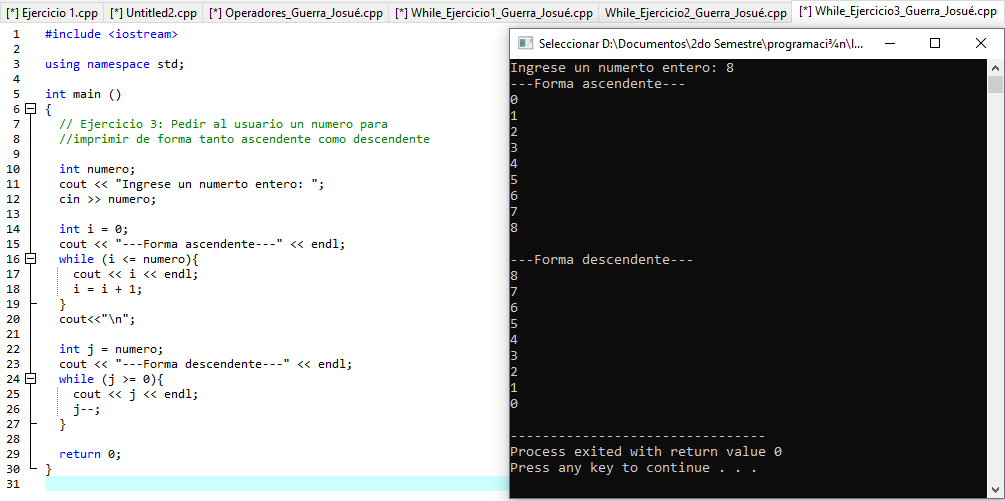
## Ejercicio 1:

## 

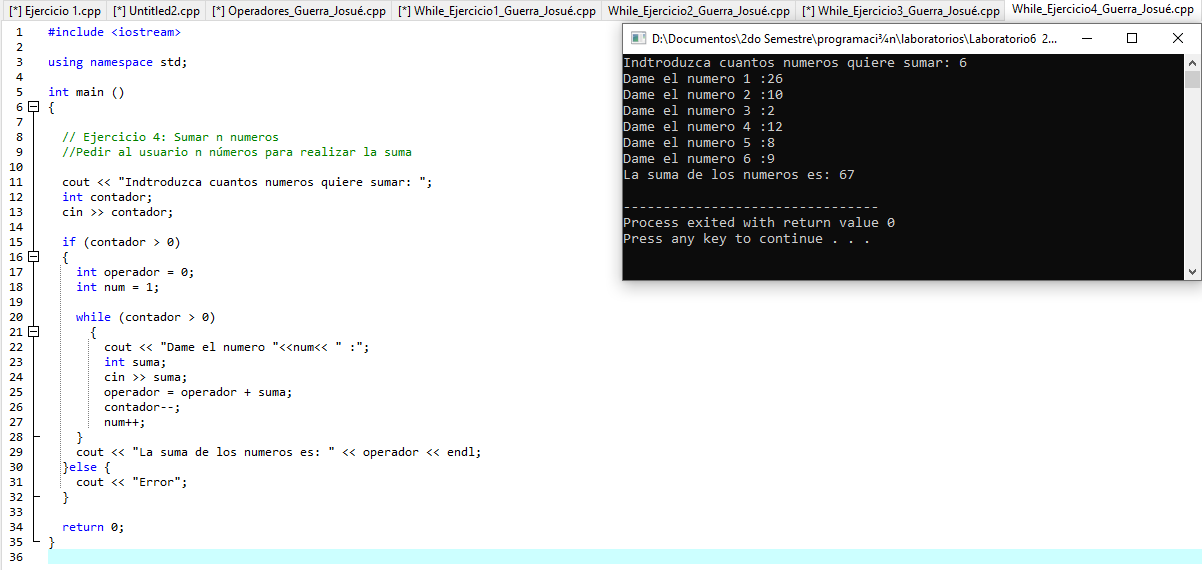
Ejercicio 2:



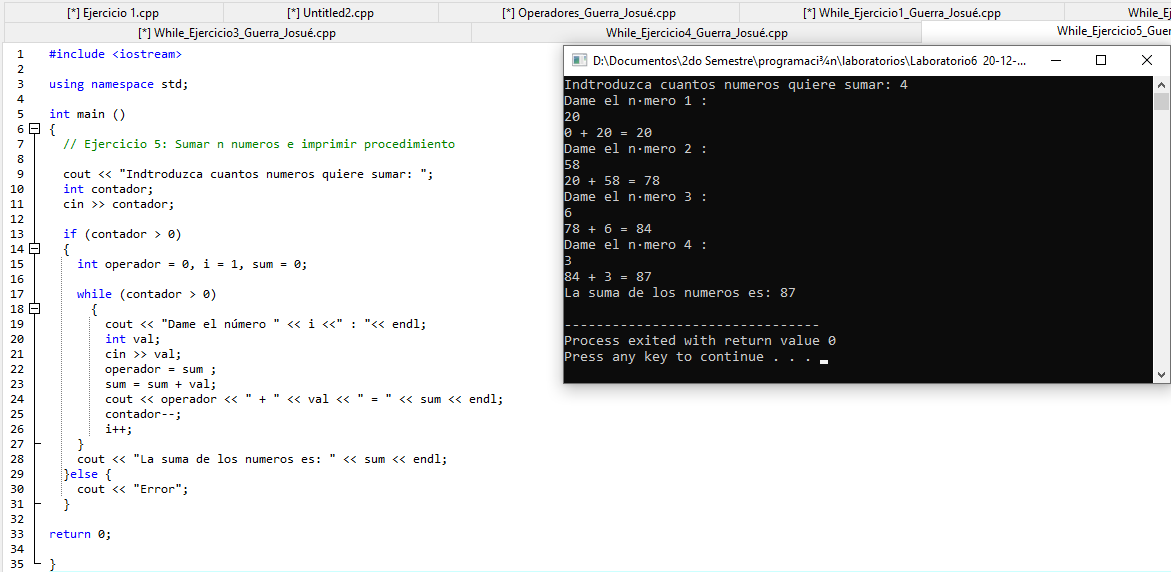
Ejercicio 3:

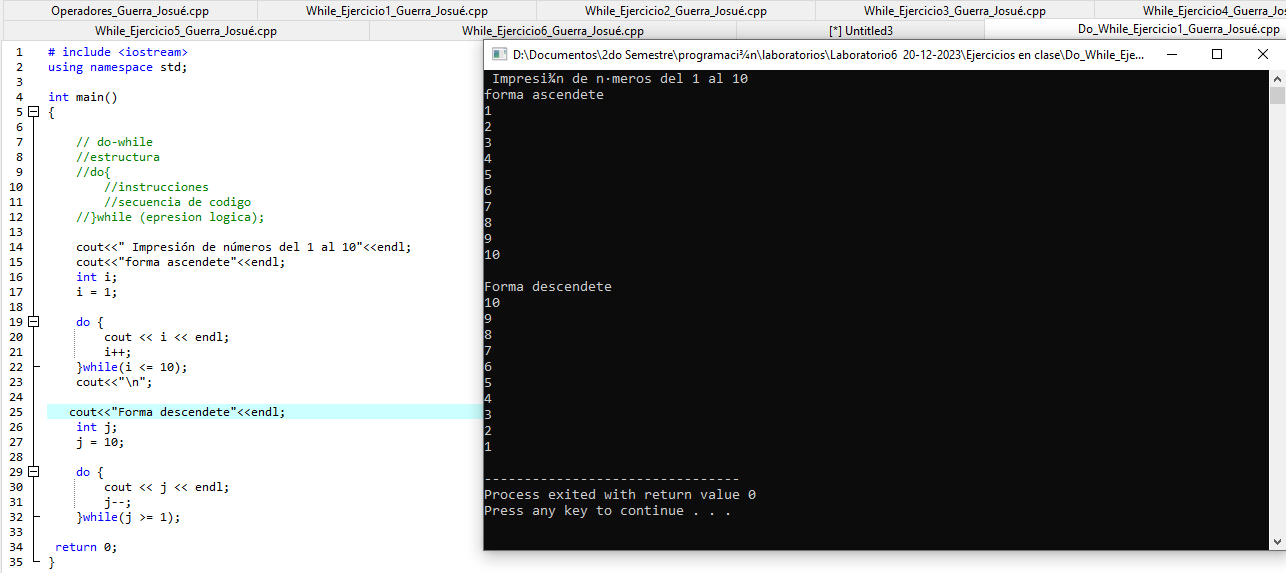
****

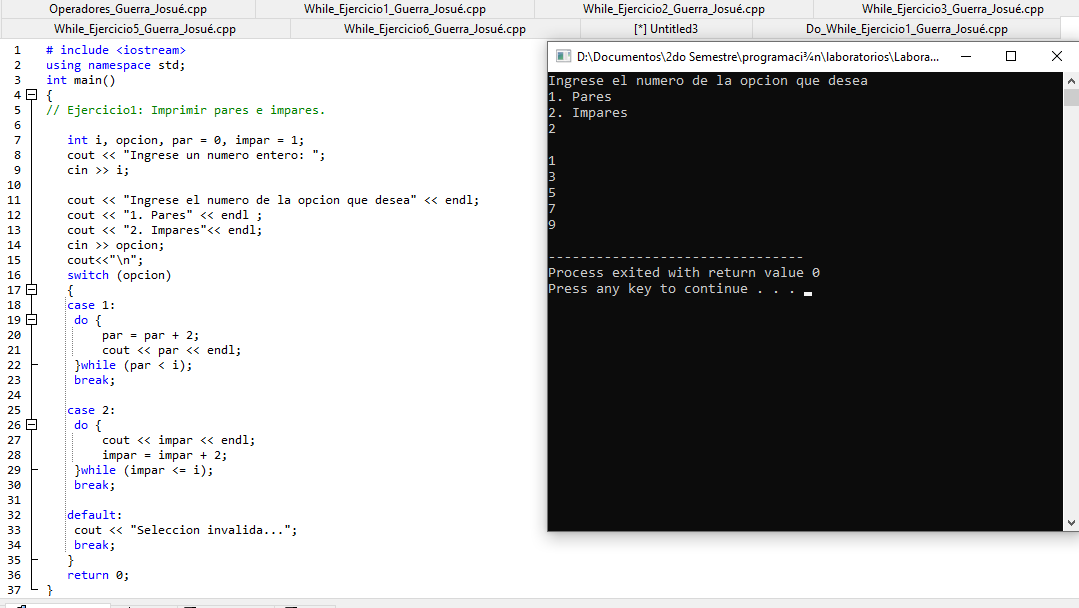
Ejercicio 4:

****

Ejercicio 5:



* + **DO-WHILE**

Ejercicio 2:

## FOR

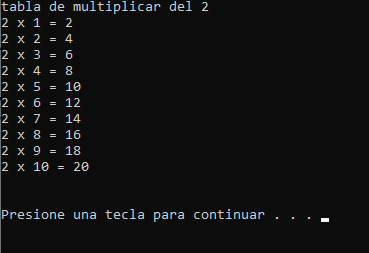
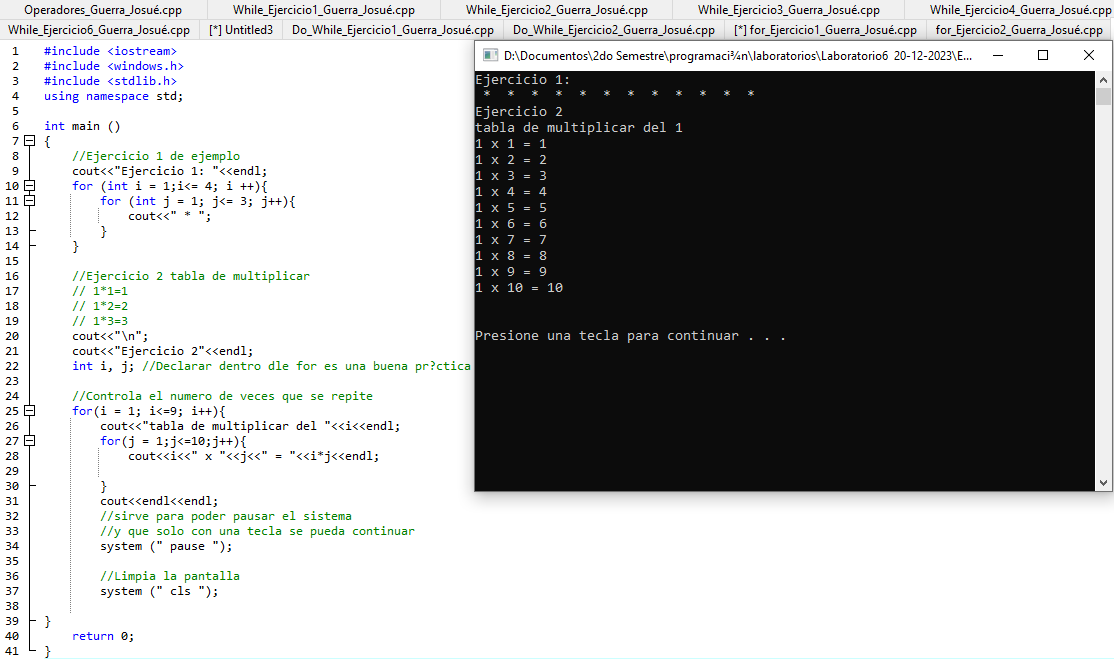
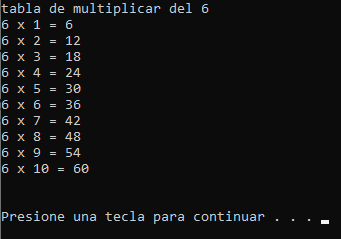
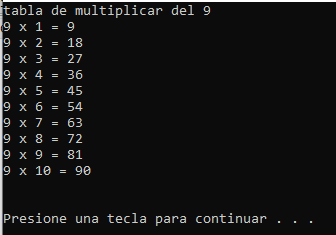
## Ejercicio 1:

## 

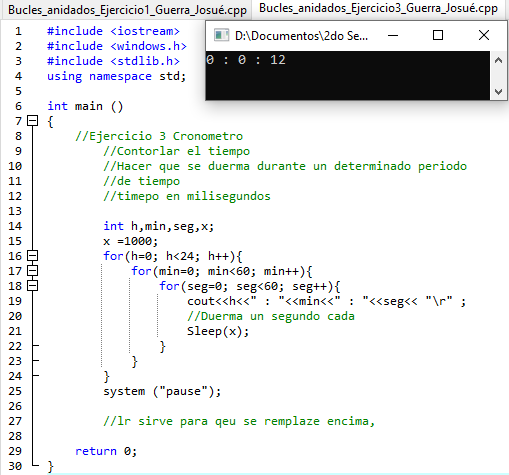
## Ejercicio 2:

## 

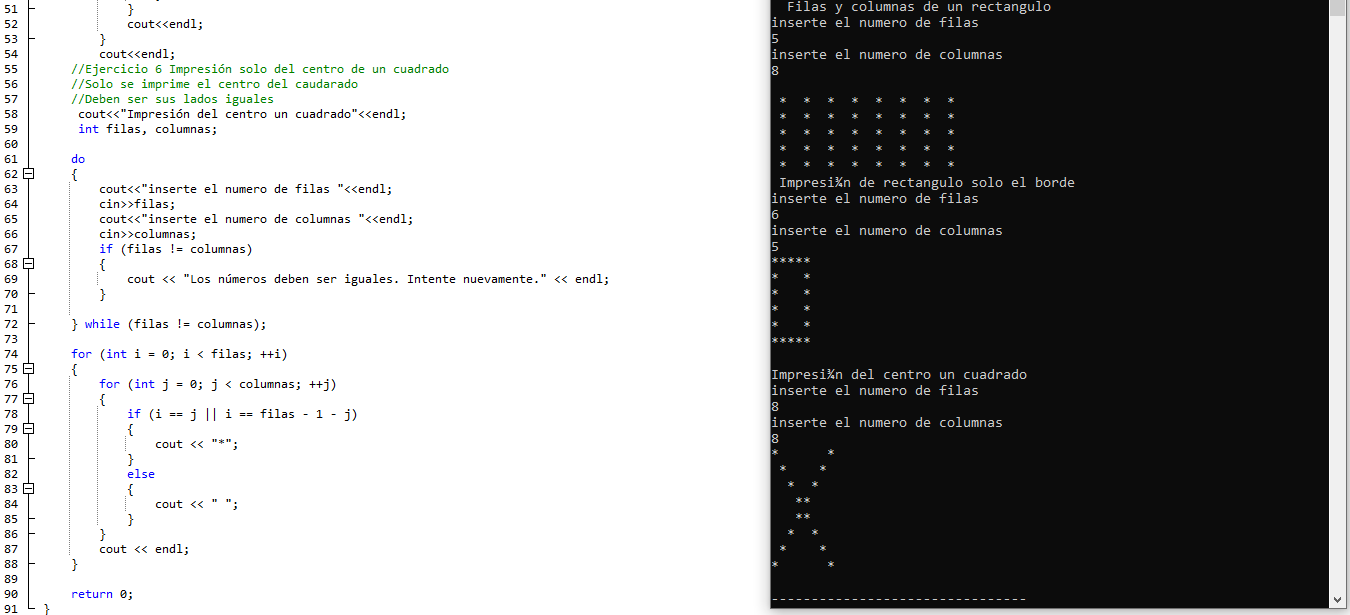
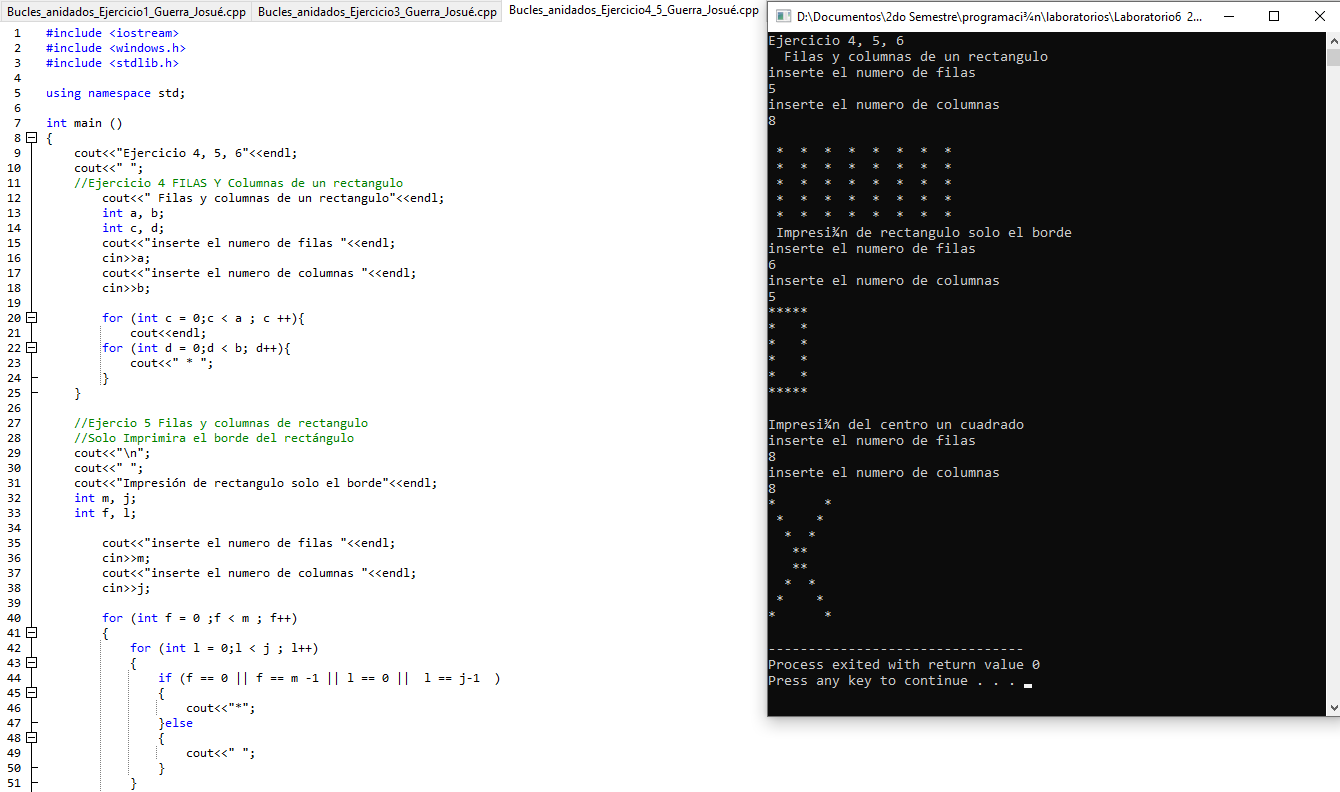
* + **BUCLES ANIDADOS**

Ejercicio 1 y 2:

El código se lo realizo de tal manera que aplastando una tecla este cambie a la siguiente tabla de multiplicar y se remplace en la pantalla, pero solo hasta la tabla del 9.

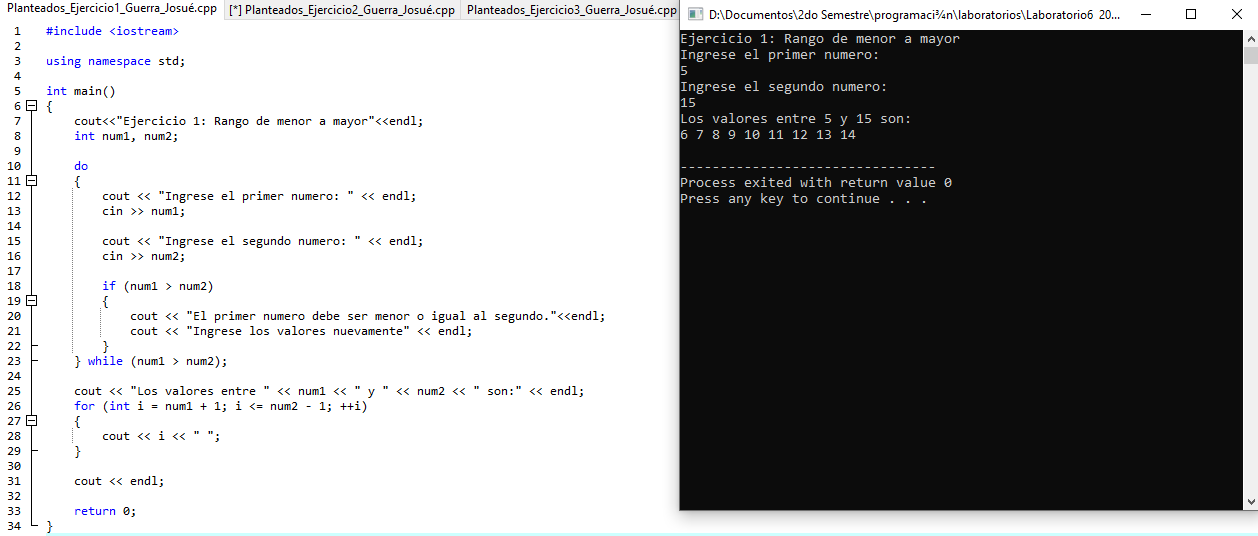
Ejercicio 3:

El código como tal muestra un cronometro, el cual en vez de ir imprimiendo los números hacia abajo solo los remplaza en pantalla.

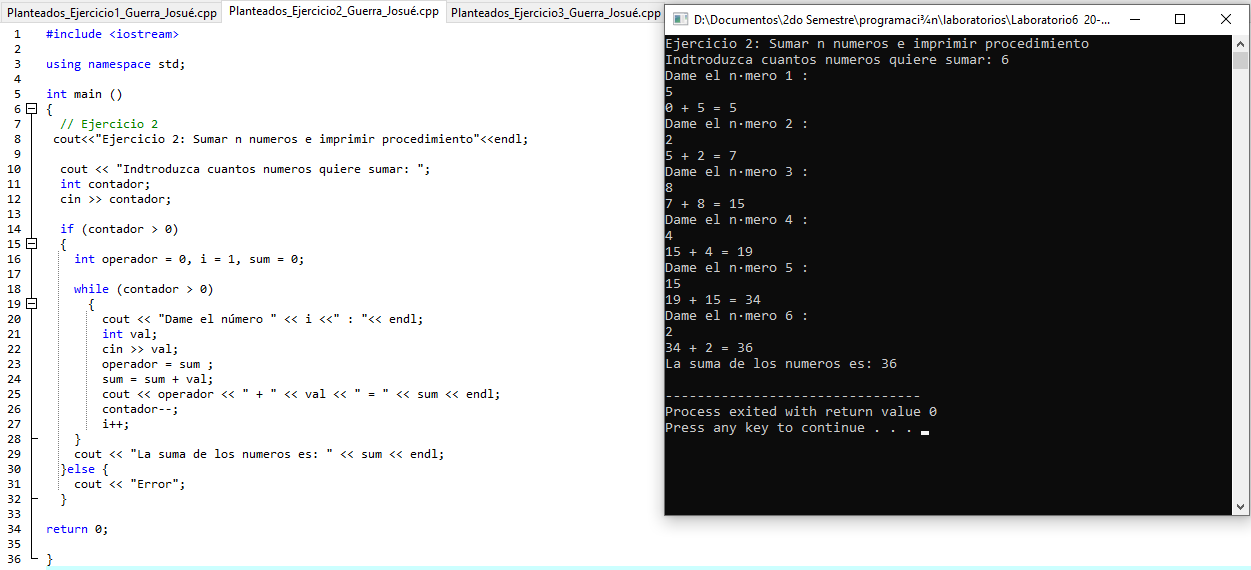
****Ejercicio 4, 5, 6:

**EJERCICIOS PLANTEADOS**

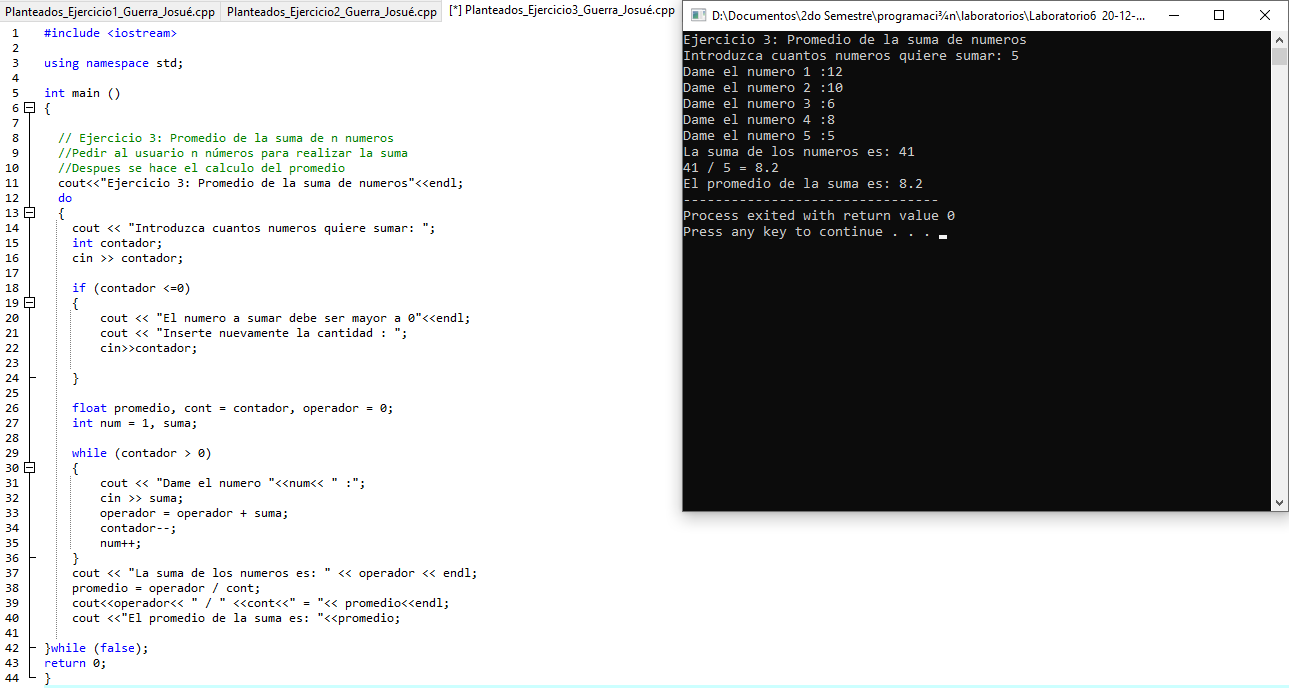
**Ejercicios planteados por la ingeniera**

* Imprimir los valores de menor al mayor existentes entre dos números.

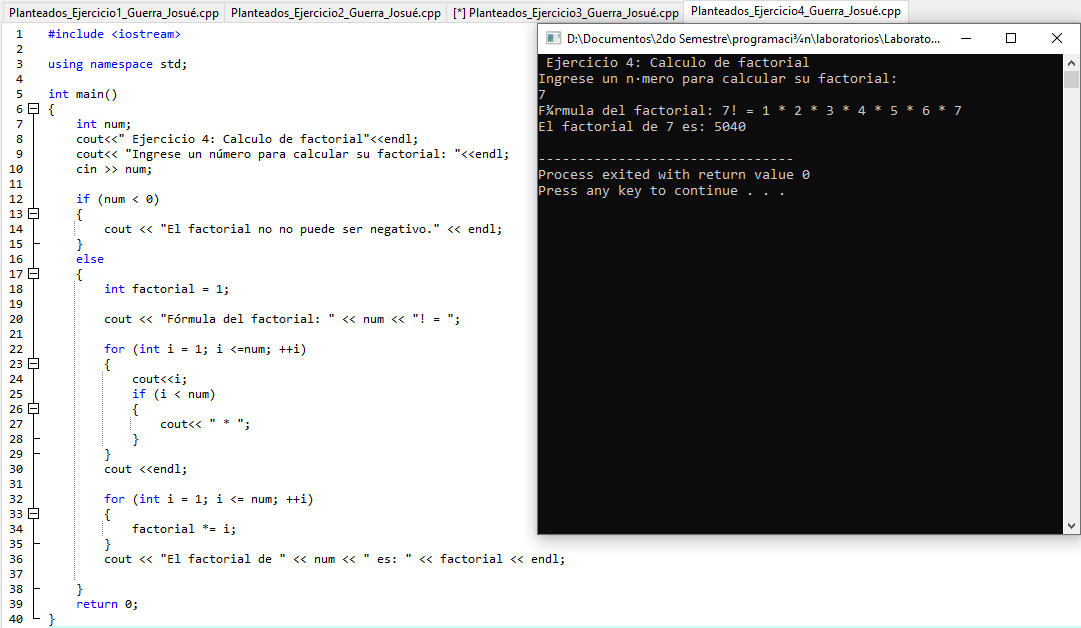
Dentro del código se solicita al usuario que inserte dos números y el programa solo muestra en pantalla desde el menar hasta el mayor sin contar los valores ingresados inicialmente.

* Sume N números ingresados por el usuario.

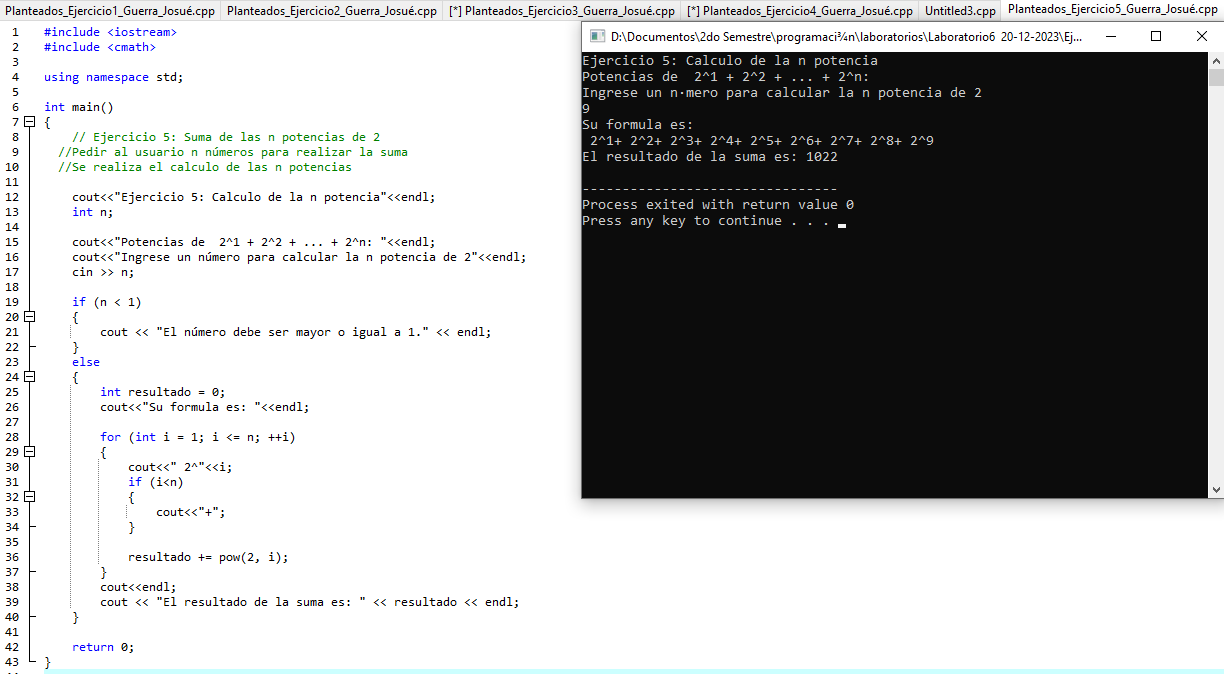
Para iniciar el programa pedirá la cantidad de números que el usuario desea sumar, el programa verificara que sea mayor que 0, una vez verificado se mostrara el procedimiento de cada operación y al final dará como resultado la suma total de todas las sumas.

* Calcule el promedio de N números dados por el usuario.

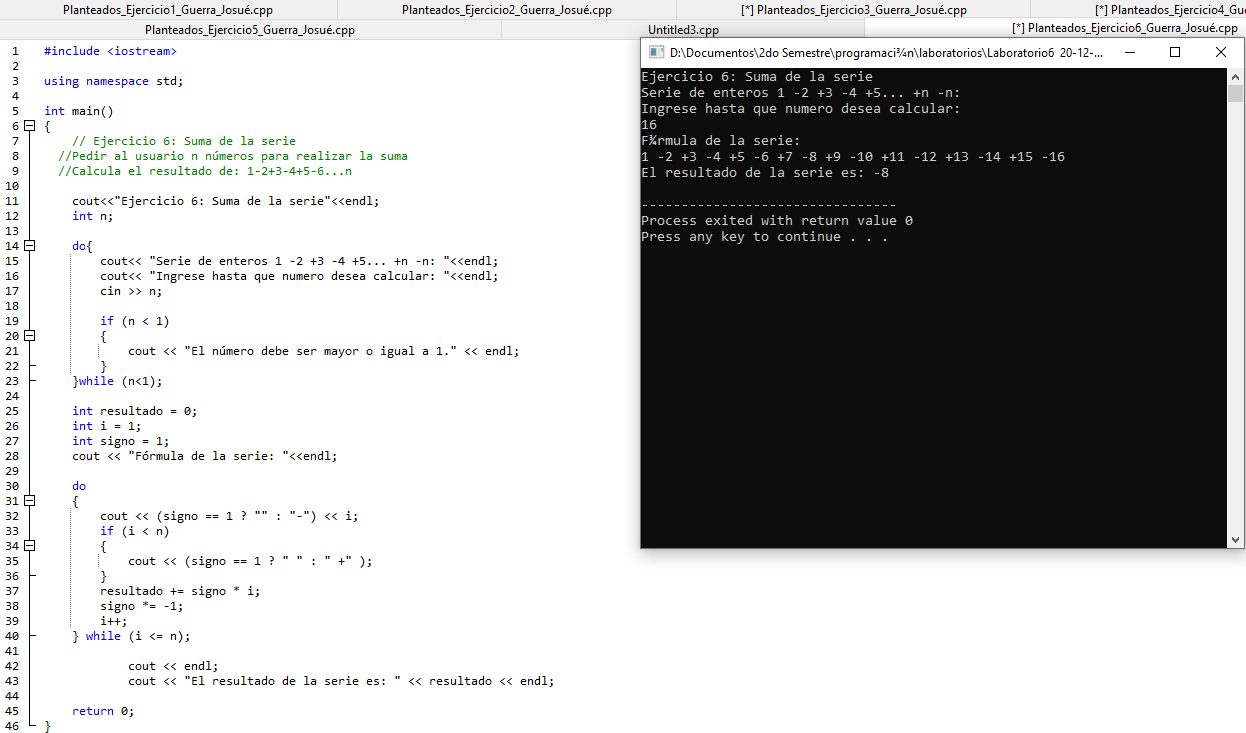
El programa solicita al usuario n cantidad de números para saber su promedio, seguido de eso solicita los números los cuales se van acumulando y al final muestra el resultado total, también dentro del código se realiza para que calcule el promedio y dicho resultado se muestre al final de todo el código en pantalla.

* Calcule la factorial de un número dado.

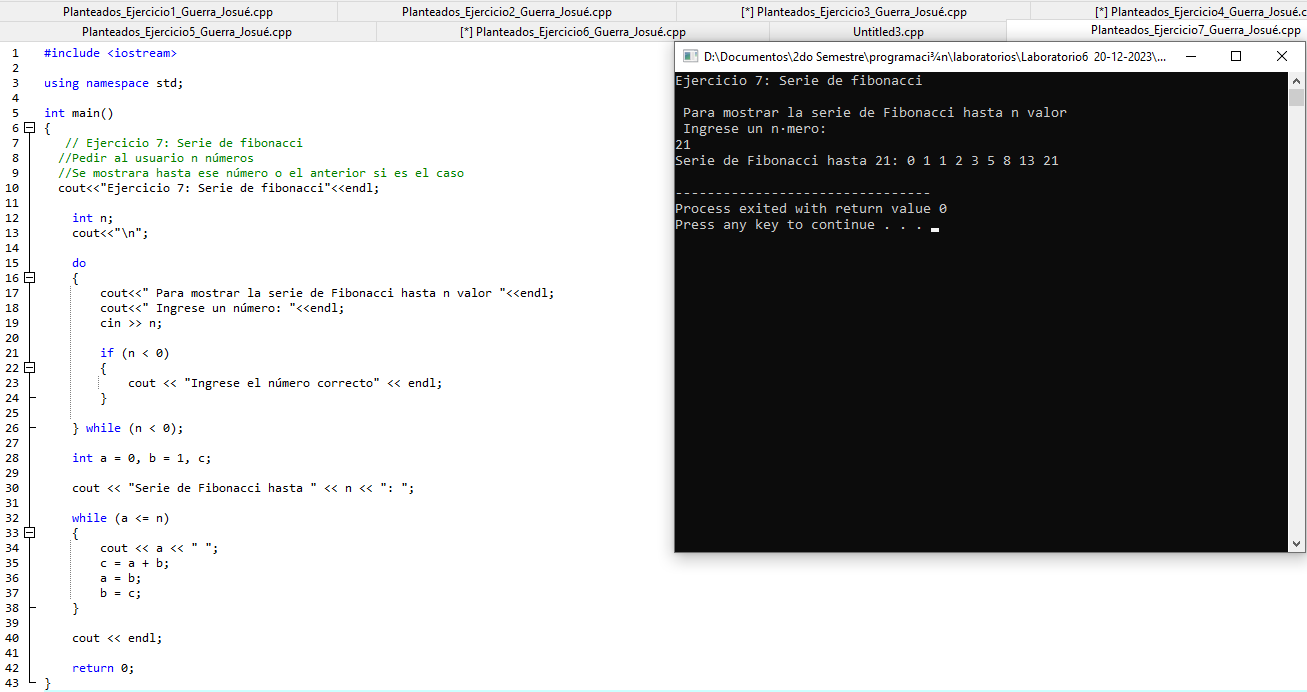
Como se sabe el factorial de un número es la suma de la multiplicación del número hasta 1, dentro del programa se realiza el cálculo y este muestra tanto la fórmula como el resultado final, obviamente se solicita al usuario que ingrese un número para realizar dicha operación.

o Calcula el valor de: 2^1+2^2+2^3+...+2^n

El problema ya viene predeterminado con el número dos como base lo que se hace es pedir al usuario que inserte un n número el cual será hasta que potencia se desea elevar y se realizara la suma respectiva de acuerdo a la potencia.

* Calcula el resultado de: 1-2+3-4+5-6...n.

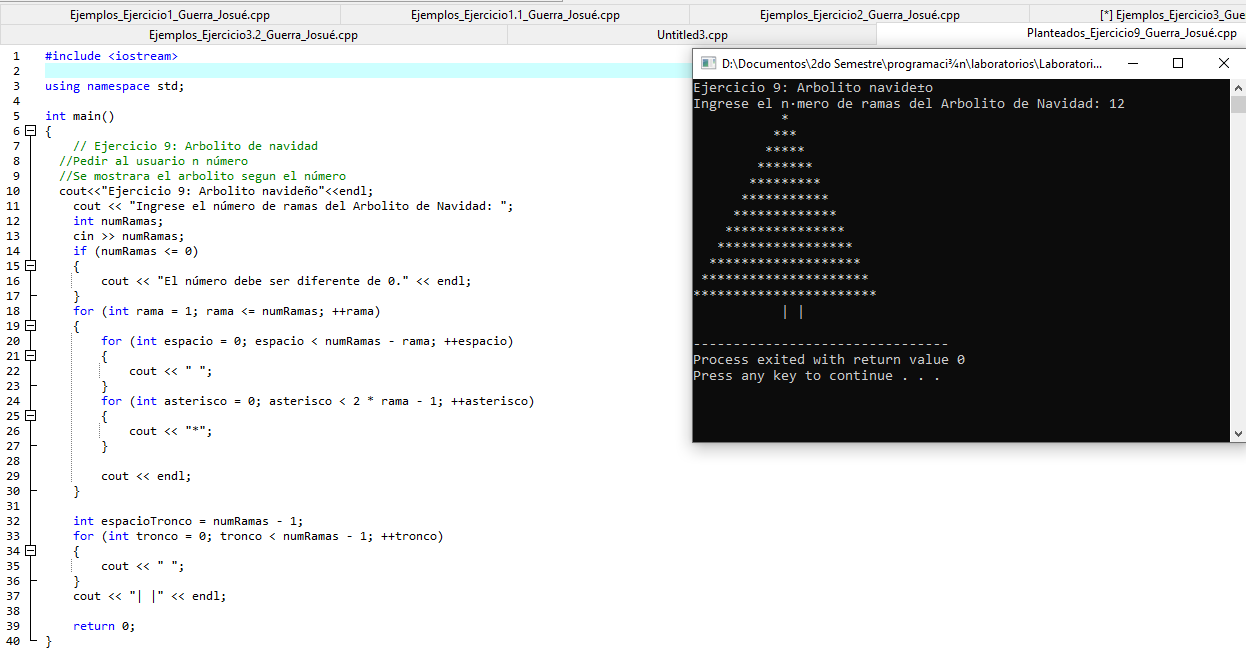
La serie planteada muestra la suma y resta de todos los números naturales, entonces en el programa ser pide al usuario que inserte un n número, el cual el programa verificara que sea mayor que 0 y realizara los distintos cálculos, al final este mostrara la respuesta total de dichas operaciones.

* Dibuje la serie Fibonacci: 1 1 2 3 5 8 13...n.
* Un número entero es un “**número perfecto**” si sus factores, incluyendo al 1 (pero excluyendo en el número mismo), suman igual que el número.

## Ejemplo:

6 es un numero perfecto porque 6= 1+2+3.

Realice un programa que diga cuales de los N números ingresados por el usuario son perfectos. 0 para salir.

* Imprima un “Arbolito de Navidad”. El Número de “ramas” debe ser dado por el usuario.

## TAREA

* Consulte y muestre otros ejemplos de funciones de las librerías:
  + CONIO.H:

Dicha librería no es considerada como parte estándar de C, sin embargo, fue utilizada para dar mejor presentación a programas, la cual fue creada por Borland, responsable del lanzamiento de múltiples compiladores de C [1].

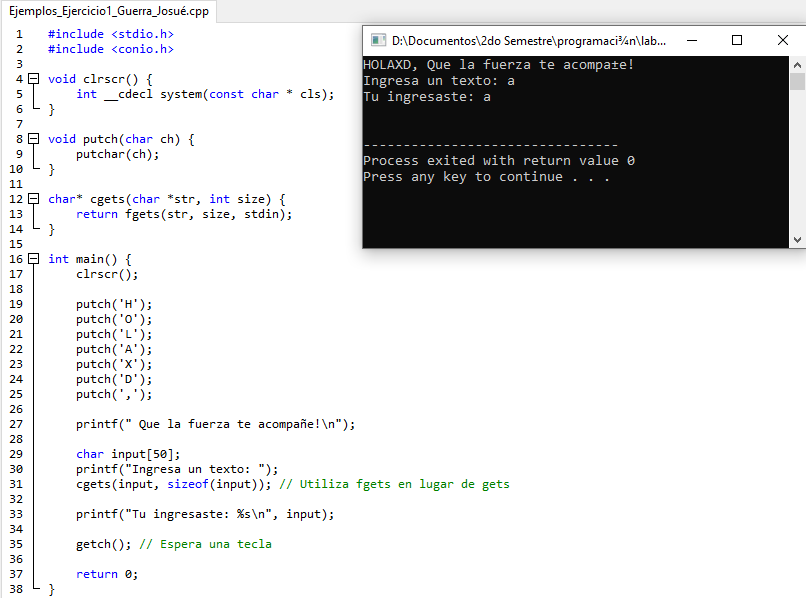
Contiene rutinas específicas de las líneas de comandos de DOS, el cual permite leer un carácter sin tener que presionar la tecla enter, es decir que la comunicación con el terminal es demasiado específica del sistema operativo [2].

Significa Console-Input-Output, contiene funciones de entrada-salida de consola utilizado principalmente por los compiladores MS-DOS [3].

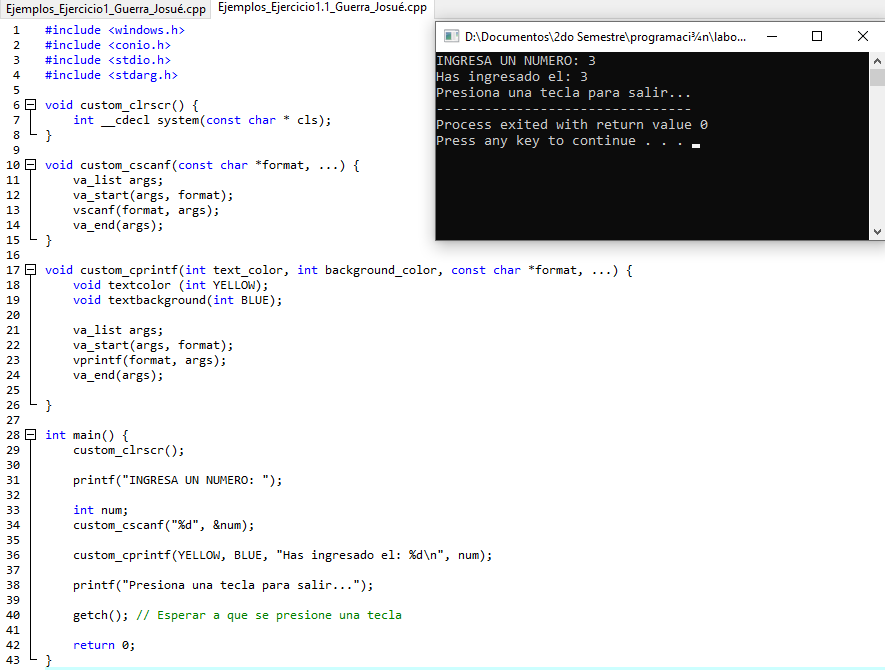
Dicha librería tiene algunas funciones las cuales son más utilizadas del encabezado conio.h las que son: [4]

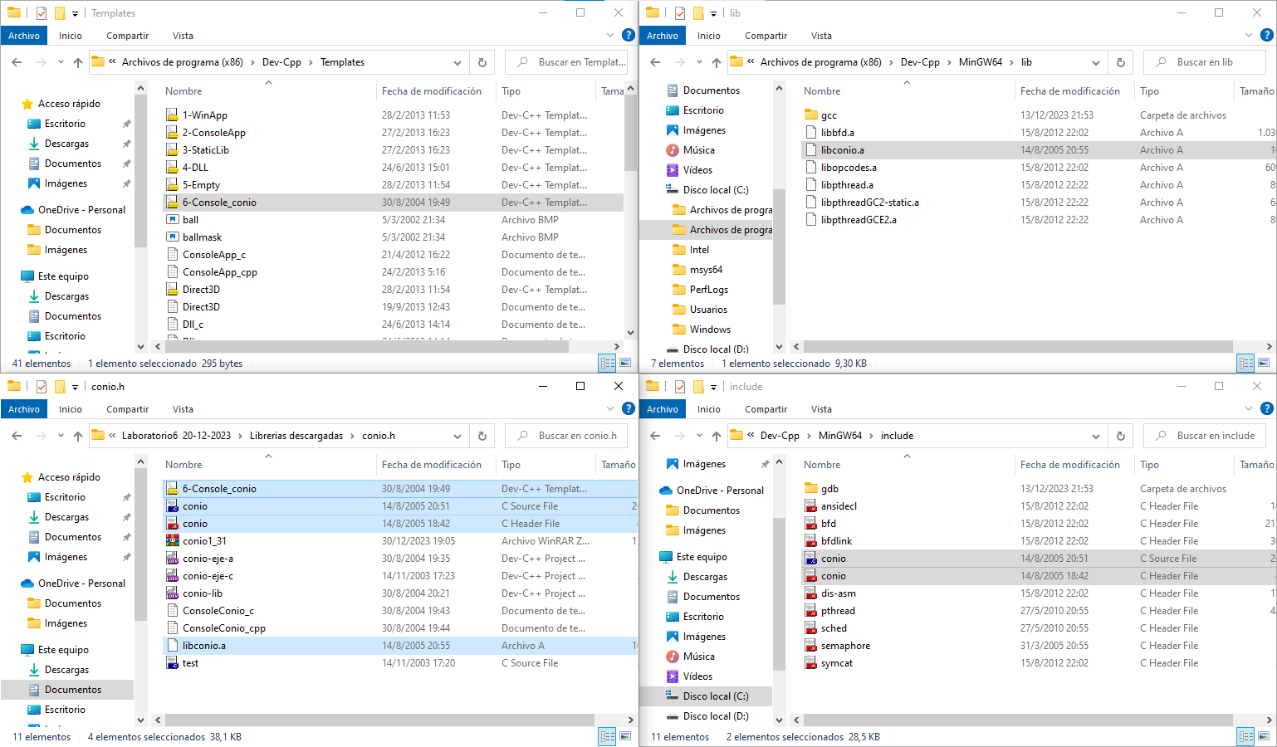
* clrscr (): Puede borrar la ventana de comandos de salida. / **void clrscr(void);**
* getch (): Se usa para mantener la pantalla de salida hasta que el usuario introduzca cualquier carácter. / **int getch(void);**
* getche (): También imprime el valor introducido por el usuario en la ventana de salida. / **char getche ();**
* putch (): Imprimir información en la ventana de salida, leyendo solo un carácter a la vez. **/int putch (int c);**
* cgets (): leer cadena de caracteres hasta que se encuentre el retorno CR y salto de línea LF, remplazando con el terminador nulo (\0). /**char\* cgets (char\* str);**
* cputs (): Caracteres de retorno de carro y de nueva línea no se anexan en la cadena convierte carácter de nueva línea (\n). /**int cputs (const char \*str);**
* cscanf (): Escanea y lee entrada de consola, para leer entrada de formato requerido, devuelve EOF cuando llega al final del flujo de entrada. / int cscanf **(char \*format, … .);**
* cprintf (): Imprime valores de salida en consola según el formato específico. /**int cprintf (const char \*format, … .);**
* kb hit (): Devuelve un valor distinto de cero cuando se ha presionado una tecla de lo contrario, devuelve cero. / **int kbhit ();**
* textcolor (): Sirve para cambiar el color de un texto. / void texcolor(int color);
* textbackground (): Sirve para cambiar el color de fondo. / void textbackground(<color>);

1. Figura 1: uso de las 5 primeras funciones [clrscr()](https://techsupportwhale.com/conio-h-c-plus-plus-library/#clrscr), [getch()](https://techsupportwhale.com/conio-h-c-plus-plus-library/#getch), [getche()](https://techsupportwhale.com/conio-h-c-plus-plus-library/#getche), [putch()](https://techsupportwhale.com/conio-h-c-plus-plus-library/#putch), [cgets()](https://techsupportwhale.com/conio-h-c-plus-plus-library/#cgets)



1. Figura 2: Uso de las siguientes 6 funciones.



Figura 3: Instalación de las librerias en DEVC++, se ve un video para realizar la respectiva instalación ya que al correr los códigos sin la librería incluida este no funcionara [5].

* + IOMANIP:

Definir varios manipuladores de los cuales cada uno tome un solo argumento, controlando

La salida [6]

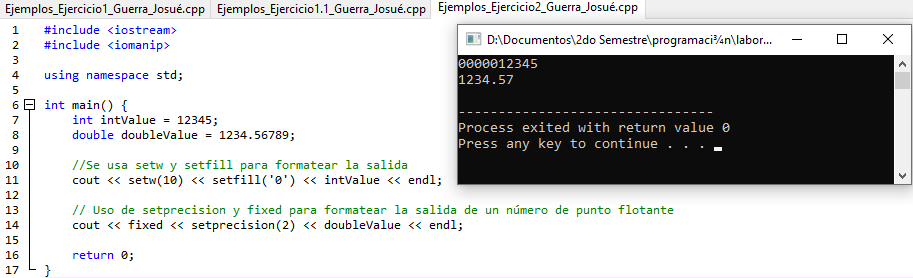
Algunas de las funciones más utilizadas son:

Setw: (Determina el espacio en la presentación)

del terminal

setfill: (Estable el carácter para rellenar espacios justificado a la derecha)

setprecision: (Precisión de los puntos flotantes)

fixed:

* + STDLIB.H

Define varias funciones, junto con la administración dinámica de la memoria, proporciona macros, definiciones de tipos y manejo de cadenas, cálculos matemáticos, incluso puede observar el espacio entre el número y texto incluyendo la cadena ptr [7], [8].

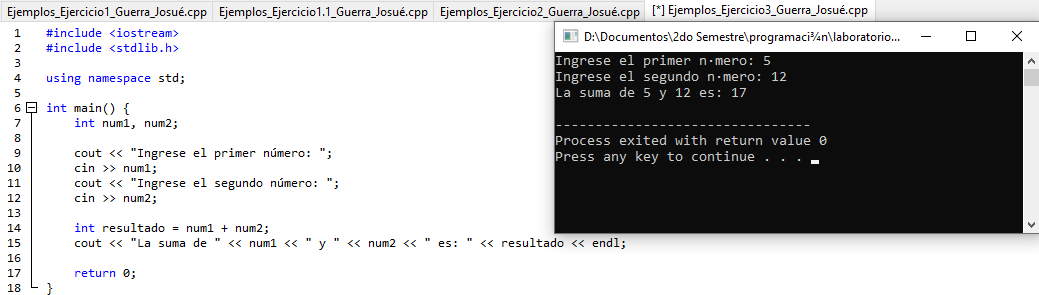
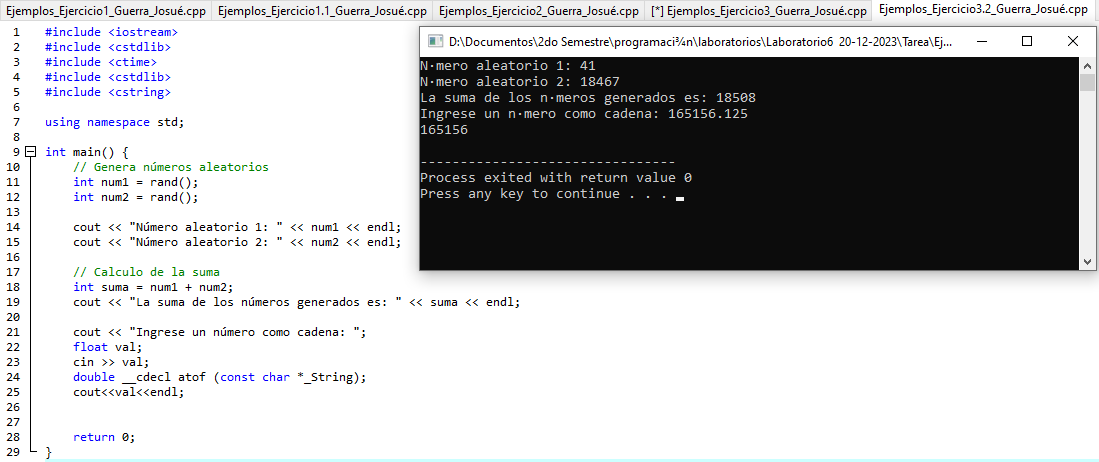
Figura 3.1

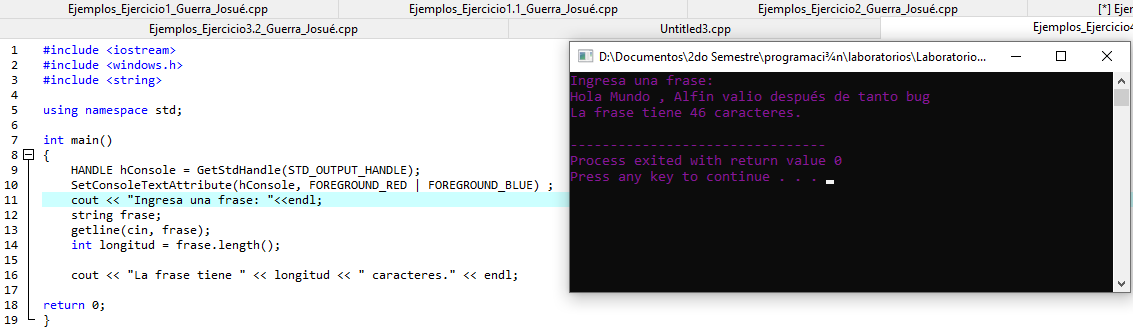
Figura 3.2:

Se usan funciones como lo son:

Rand: para creación de números aleatorios.

Atof: para convertir una cadena en doble.

* + WINDOWS.H:

  
Contiene las definiciones de todas las funciones de la biblioteca Windows API, así como las macros utilizadas por los desarrolladores de aplicaciones para Windows. Además, incluye las estructuras de datos empleadas en numerosas funciones y subsistemas dentro del entorno Windows, uno de las más características más usadas es el system pause [9].

## PRESENTACIÓN

Al finalizar tu laboratorio deberás subir un archivo en formato pdf con el nombre (Laboratorio6\_Programación\_NApellido)

## CONCLUSIONES:

## Se sabe bien que las estructuras iterativas como repetitivas cumplen funciones importantes dentro de los códigos de programación, el tener buen conocimiento y aplicación de estructuras iterativas, como los bucles for, while, y do-while, nos ayuda a poder abordar una variedad de problemas en situaciones cotidianas.

## A través de la práctica y desarrollo de ejercicios, es posible adquirir una comprensión más profunda y clara de cómo estas estructuras pueden ser empleadas de manera efectiva.

## En conclusión, comprender y aplicar las estructuras iterativas en contextos cotidianos no solo fortalece las habilidades de programación, sino que también mejora la capacidad para abordar una amplia gama de desafíos en el mundo real. La práctica constante y el desarrollo de ejercicios son clave para internalizar estos conceptos y utilizarlos de manera efectiva en la resolución de problemas. Tomando en cuenta el uso de las diferentes bibliotecas que se pueden implementar dentro de los códigos para facilitar algunos procesos.

## RECOMENDACIONES:

Es recomendable poder fortalecer las habilidades de programación mediante la comprensión y aplicación de estructuras iterativas es desarrollar paciencia y perseverancia en la práctica diaria. La programación, como cualquier otra habilidad, se perfecciona con el tiempo y la repetición. Dedicar tiempo regularmente a resolver problemas, desarrollar proyectos y explorar diferentes usos de los bucles for, while y do- while le ayudará no sólo a adquirir conocimientos técnicos, sino también a internalizar los conceptos subyacentes más profundamente. Además, es recomendable no subestime el valor de colaborar con la comunidad de programación, participar en proyectos conjuntos y aprender de experiencias compartidas. Interactuar con otros desarrolladores puede enriquecer su desarrollo como programador, brindándole nuevas perspectivas, enfoques innovadores e intercambio de conocimientos valiosos.

## ENLACES

* En este apartado se presentan los enlaces donde se encuentra los códigos de cada ejercicio.
* Enlace GITHUB:

Prof. Ing. Ivonne Maldonado 1