



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



PROGRAMACION TDSD214

ASIGNATURA:

Programación

PROFESOR:

Ing. Ivonne Maldonado

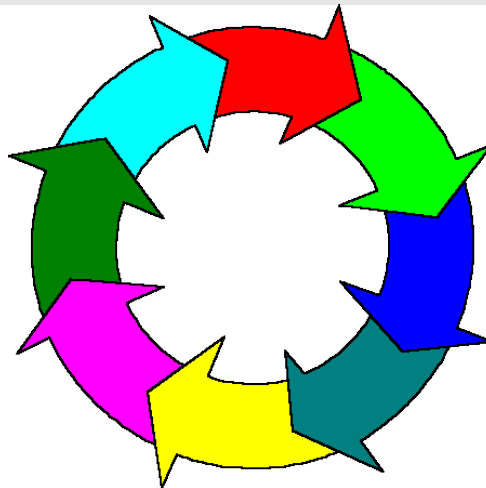
PERÍODO ACADÉMICO:

2023-B

LABORATORIO - 6

TÍTULO:

ESTRUCTURAS ITERATIVAS



Nombre:

Guerra Lovato Josué Eduard

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.

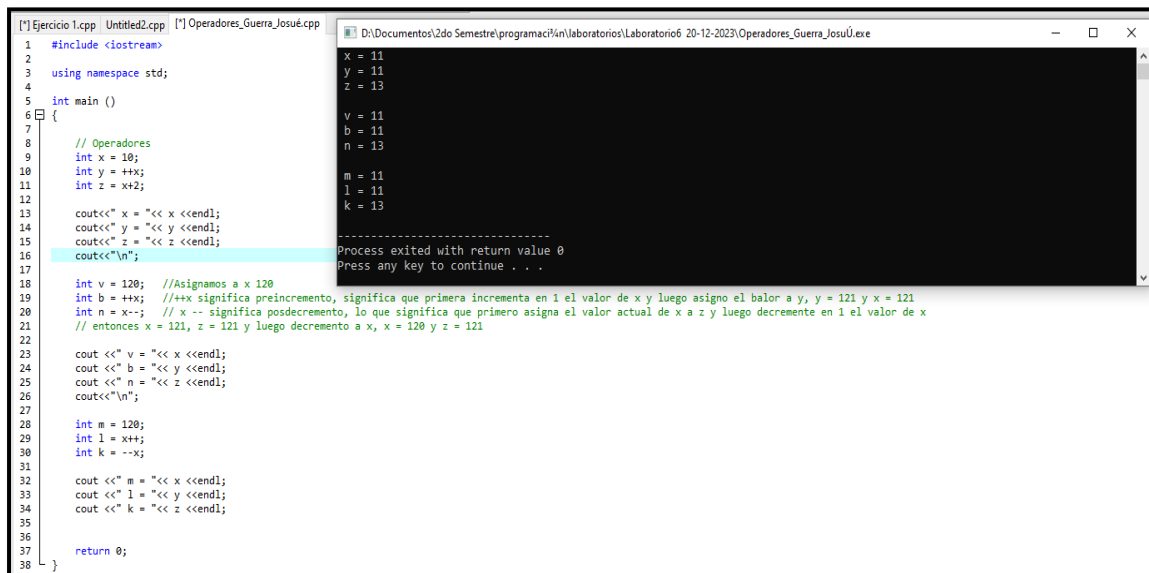
OBJETIVOS ESPECÍFICOS

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

INSTRUCCIONES

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

• OPERADORES



```
[*] Ejercicio 1.cpp  Untitled2.cpp  [*] Operadores_Guerra_Josué.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main ()
6  {
7
8      // Operadores
9      int x = 10;
10     int y = ++x;
11     int z = x+2;
12
13     cout<<" x = "<< x <<endl;
14     cout<<" y = "<< y <<endl;
15     cout<<" z = "<< z <<endl;
16     cout<<"\n";
17
18     int v = 120; //Asignamos a x 120
19     int b = ++x; //++x significa preincremento, significa que primera incrementa en 1 el valor de x y luego asigno el valor a y, y = 121 y x = 121
20     int n = x--; // x -- significa posdecremento, lo que significa que primero asigna el valor actual de x a z y luego decremente en 1 el valor de x
21     // entonces x = 121, z = 121 y luego decremento a x, x = 120 y z = 121
22
23     cout <<" v = "<< x <<endl;
24     cout <<" b = "<< y <<endl;
25     cout <<" n = "<< z <<endl;
26     cout<<"\n";
27
28     int m = 120;
29     int l = x++;
30     int k = --x;
31
32     cout <<" m = "<< x <<endl;
33     cout <<" l = "<< y <<endl;
34     cout <<" k = "<< z <<endl;
35
36
37     return 0;
38 }
```

```
D:\Documentos\2do Semestre\programaci\laboratorio6 20-12-2023\Operadores_Guerra_Josué.exe
x = 11
y = 11
z = 13

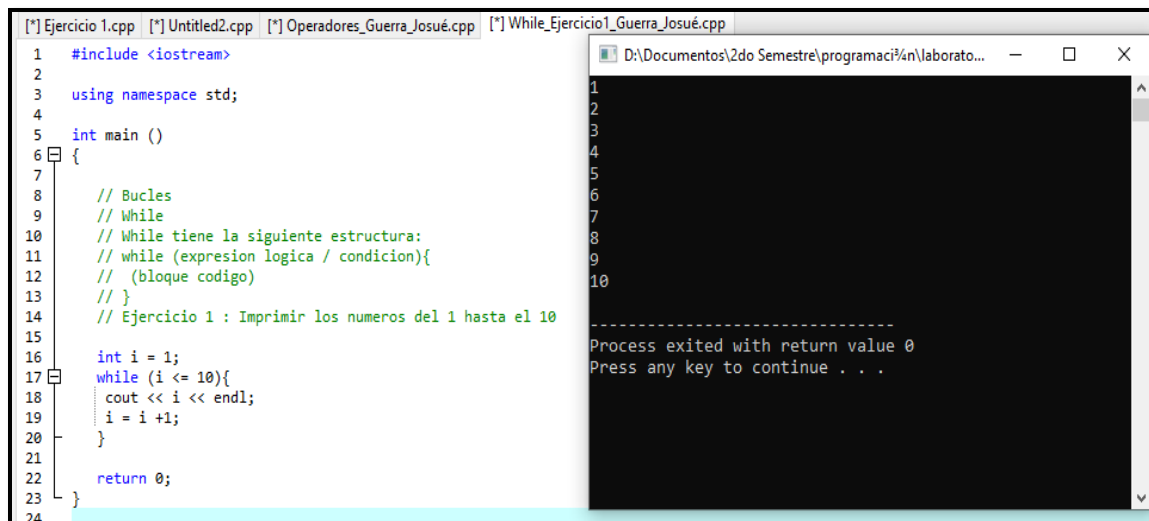
v = 11
b = 11
n = 13

m = 11
l = 11
k = 13

-----
Process exited with return value 0
Press any key to continue . . .
```

• WHILE

Ejercicio 1:

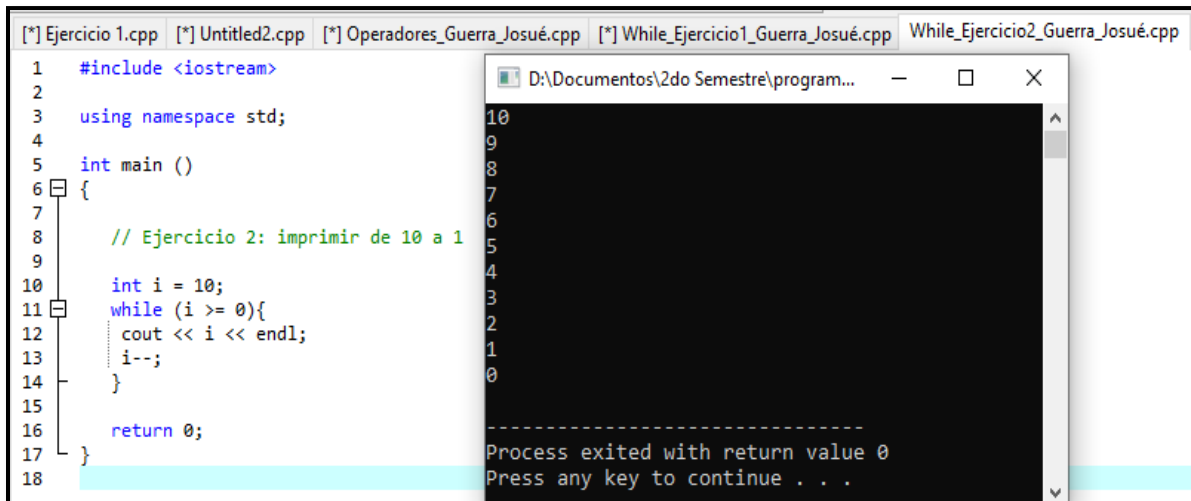


```
[*] Ejercicio 1.cpp  [*] Untitled2.cpp  [*] Operadores_Guerra_Josué.cpp  [*] While_Ejercicio1_Guerra_Josué.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main ()
6  {
7
8      // Bucles
9      // While
10     // While tiene la siguiente estructura:
11     // while (expresion logica / condicion){
12     //     (bloque codigo)
13     // }
14     // Ejercicio 1 : Imprimir los numeros del 1 hasta el 10
15
16     int i = 1;
17     while (i <= 10){
18         cout << i << endl;
19         i = i +1;
20     }
21
22     return 0;
23 }
24
```

```
D:\Documentos\2do Semestre\programaci\laborato...
1
2
3
4
5
6
7
8
9
10

-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicio 2:



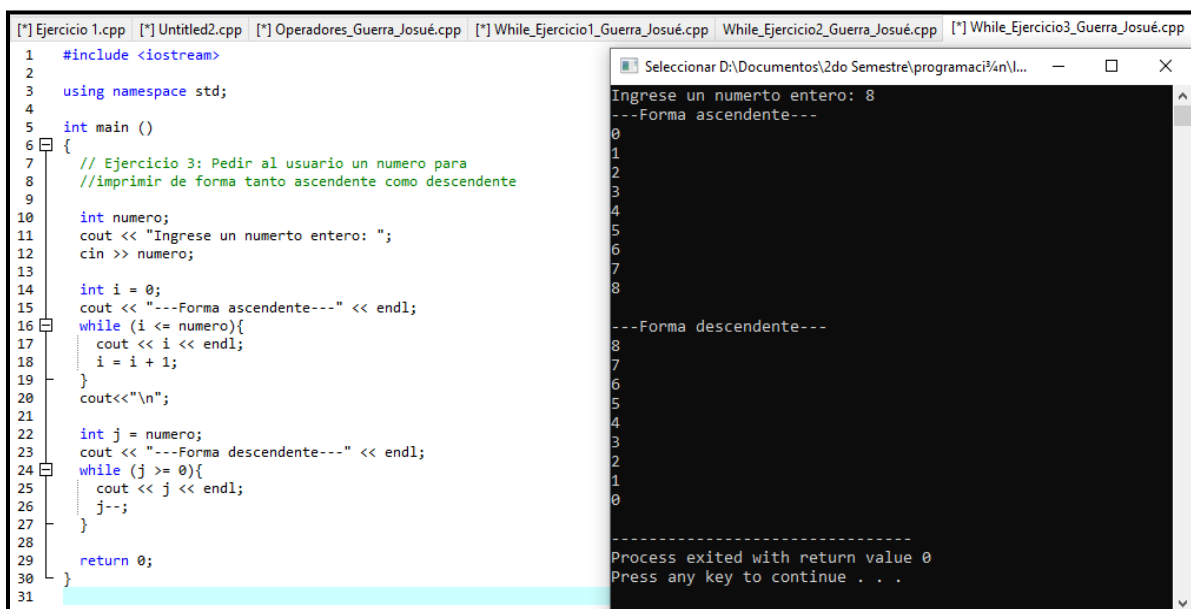
The screenshot shows a C++ IDE with the following code in `While_Ejercicio2_Guerra_Josué.cpp`:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main ()
6 {
7     // Ejercicio 2: imprimir de 10 a 1
8
9     int i = 10;
10    while (i >= 0){
11        cout << i << endl;
12        i--;
13    }
14
15    return 0;
16 }
17
18
```

The execution window shows the output of the program:

```
10
9
8
7
6
5
4
3
2
1
0
-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicio 3:



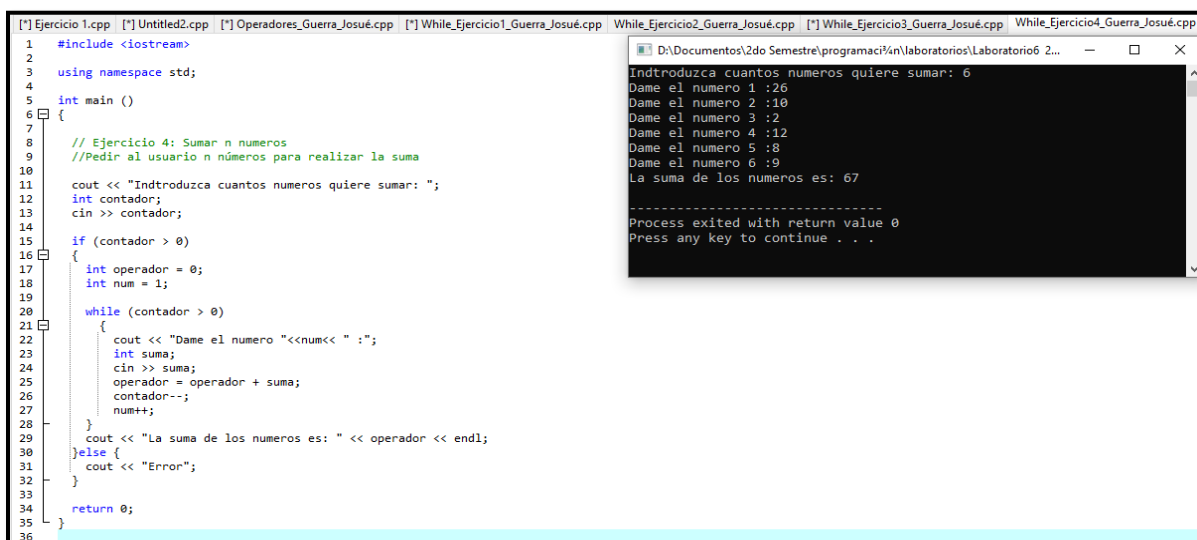
The screenshot shows a C++ IDE with the following code in `While_Ejercicio3_Guerra_Josué.cpp`:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main ()
6 {
7     // Ejercicio 3: Pedir al usuario un numero para
8     //imprimir de forma tanto ascendente como descendente
9
10    int numero;
11    cout << "Ingrese un numerto entero: ";
12    cin >> numero;
13
14    int i = 0;
15    cout << "---Forma ascendente---" << endl;
16    while (i <= numero){
17        cout << i << endl;
18        i = i + 1;
19    }
20    cout<<"\n";
21
22    int j = numero;
23    cout << "---Forma descendente---" << endl;
24    while (j >= 0){
25        cout << j << endl;
26        j--;
27    }
28
29    return 0;
30 }
31
```

The execution window shows the output of the program:

```
Ingrese un numerto entero: 8
---Forma ascendente---
0
1
2
3
4
5
6
7
8
---Forma descendente---
8
7
6
5
4
3
2
1
0
-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicio 4:



The screenshot shows a C++ IDE with the following code in `While_Ejercicio4_Guerra_Josué.cpp`:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main ()
6 {
7     // Ejercicio 4: Sumar n numeros
8     //Pedir al usuario n números para realizar la suma
9
10    cout << "Introduzca cuantos numeros quiere sumar: ";
11    int contador;
12    cin >> contador;
13
14    if (contador > 0)
15    {
16        int operador = 0;
17        int num = 1;
18
19        while (contador > 0)
20        {
21            cout << "Dame el numero "<<num<< " :";
22            int suma;
23            cin >> suma;
24            operador = operador + suma;
25            contador--;
26            num++;
27        }
28        cout << "La suma de los numeros es: " << operador << endl;
29    }
30    else {
31        cout << "Error";
32    }
33
34    return 0;
35 }
36
```

The execution window shows the output of the program:

```
Introduzca cuantos numeros quiere sumar: 6
Dame el numero 1 :26
Dame el numero 2 :10
Dame el numero 3 :2
Dame el numero 4 :12
Dame el numero 5 :8
Dame el numero 6 :9
La suma de los numeros es: 67
-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicio 5:

The screenshot shows a C++ IDE with two windows. The left window displays the source code for 'Ejercicio 5', which uses a while loop to sum numbers from 1 to a user-defined value. The right window shows the program's execution output.

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

int main ()
{
    // Ejercicio 5: Sumar n numeros e imprimir procedimiento
    cout << "Introduzca cuantos numeros quiere sumar: ";
    int contador;
    cin >> contador;

    if (contador > 0)
    {
        int operador = 0, i = 1, sum = 0;

        while (contador > 0)
        {
            cout << "Dame el número " << i << " : "<< endl;
            int val;
            cin >> val;
            operador = sum ;
            sum = sum + val;
            cout << operador << " + " << val << " = " << sum << endl;
            contador--;
            i++;
        }
        cout << "La suma de los numeros es: " << sum << endl;
    }else {
        cout << "Error";
    }

    return 0;
}
```

Output window content:

```
Introduzca cuantos numeros quiere sumar: 4
Dame el n.mero 1 :
20
0 + 20 = 20
Dame el n.mero 2 :
58
20 + 58 = 78
Dame el n.mero 3 :
6
78 + 6 = 84
Dame el n.mero 4 :
3
84 + 3 = 87
La suma de los numeros es: 87

-----
Process exited with return value 0
Press any key to continue . . .
```

• DO-WHILE

The screenshot shows a C++ IDE with two windows. The left window displays the source code for 'Do-While Ejercicio 1', which uses do-while loops to print numbers from 1 to 10 in ascending and descending order. The right window shows the program's execution output.

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

int main()
{
    // do-while
    //estructura
    //do{
    //instrucciones
    //secuencia de codigo
    //}while (epresion logica);

    cout<<" Impresión de números del 1 al 10"<<endl;
    cout<<"Forma ascendete"<<endl;
    int i;
    i = 1;

    do {
        cout << i << endl;
        i++;
    }while(i <= 10);
    cout<<"\n";

    cout<<"Forma descendete"<<endl;
    int j;
    j = 10;

    do {
        cout << j << endl;
        j--;
    }while(j >= 1);

    return 0;
}
```

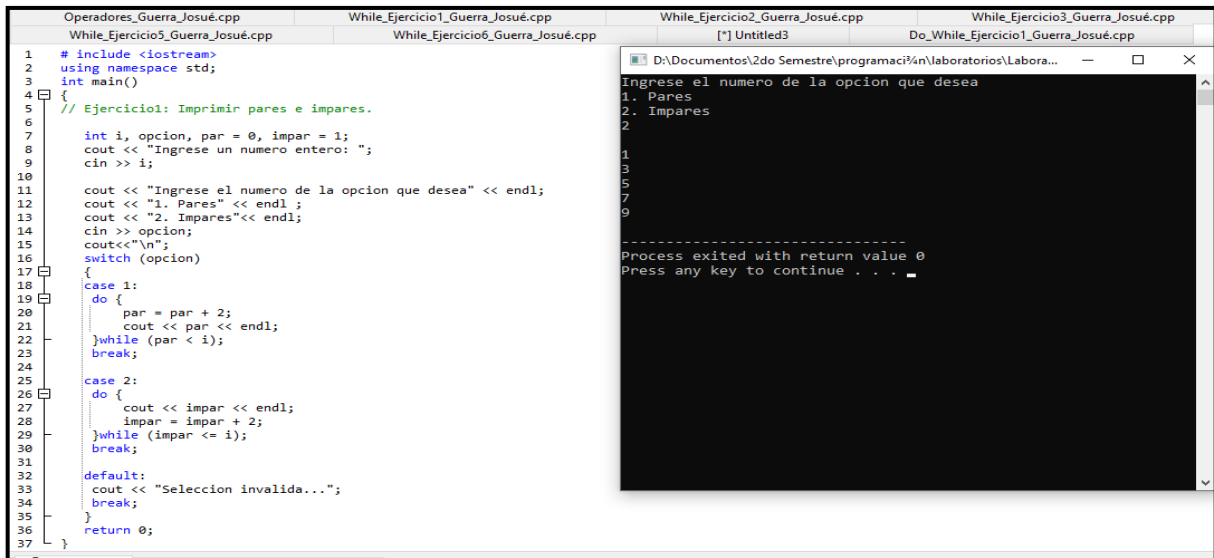
Output window content:

```
Impresión de n.meros del 1 al 10
Forma ascendete
1
2
3
4
5
6
7
8
9
10

Forma descendete
10
9
8
7
6
5
4
3
2
1

-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicio 2:



The screenshot shows a C++ IDE with several tabs. The active tab is 'Operadores_Guerra_Josué.cpp', which contains the following code:

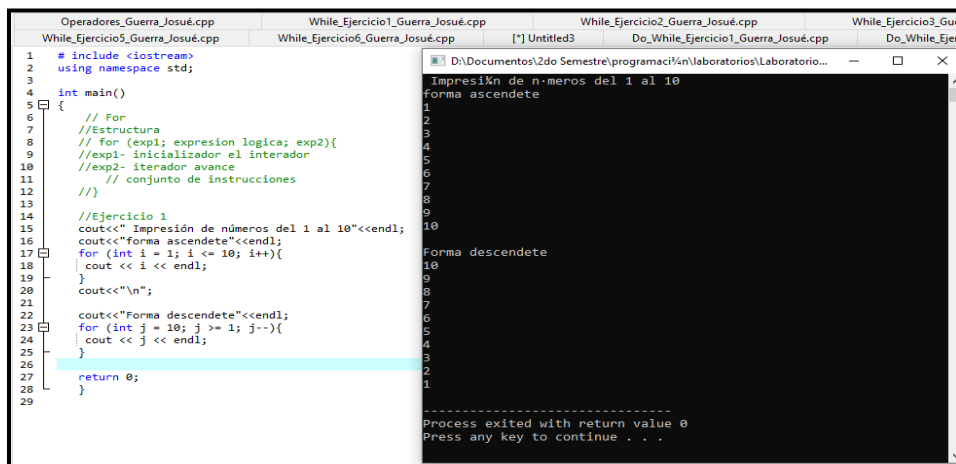
```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     // Ejercicio1: Imprimir pares e impares.
6
7     int i, opcion, par = 0, impar = 1;
8     cout << "Ingrese un numero entero: ";
9     cin >> i;
10
11     cout << "Ingrese el numero de la opcion que desea" << endl;
12     cout << "1. Pares" << endl;
13     cout << "2. Impares" << endl;
14     cin >> opcion;
15     cout << "\n";
16     switch (opcion)
17     {
18     case 1:
19     do {
20         par = par + 2;
21         cout << par << endl;
22     }while (par < i);
23     break;
24
25     case 2:
26     do {
27         cout << impar << endl;
28         impar = impar + 2;
29     }while (impar <= i);
30     break;
31
32     default:
33     cout << "Seleccion invalida...";
34     break;
35
36     return 0;
37 }
```

The output window shows the following text:

```
D:\Documentos\2do Semestre\programaci4n\laboratorios\Labora...
1. Pares
2. Impares
2
1
3
5
7
9
-----
Process exited with return value 0
Press any key to continue . . .
```

• FOR

Ejercicio 1:



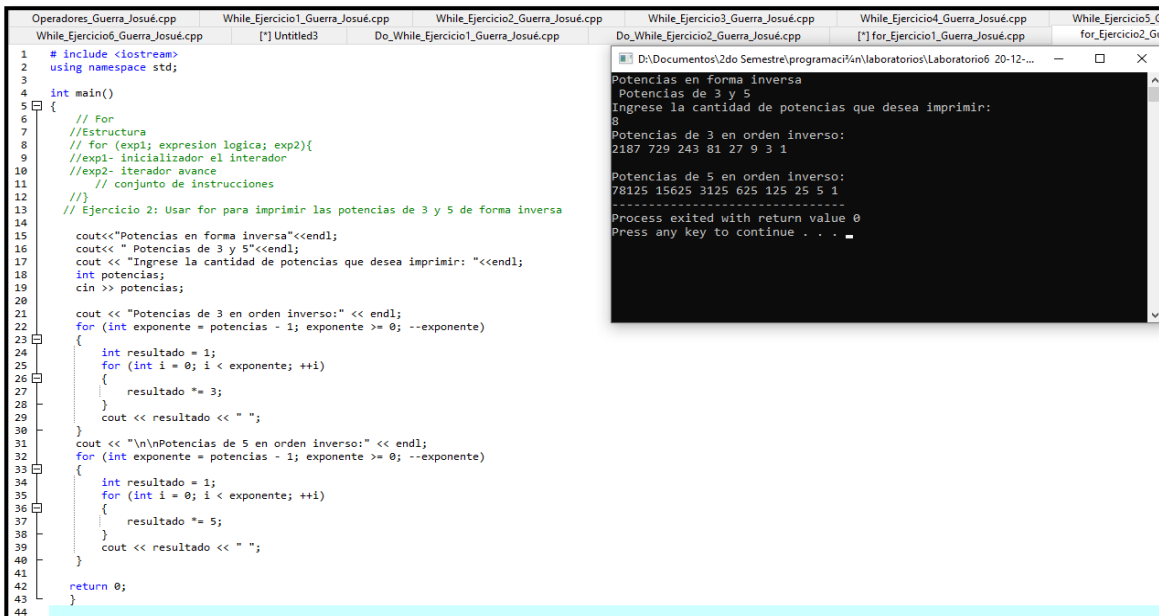
The screenshot shows a C++ IDE with several tabs. The active tab is 'Operadores_Guerra_Josué.cpp', which contains the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     // For
7     //Estructura
8     // for (exp1; expresion logica; exp2){
9     //exp1- inicializador el interador
10    //exp2- iterador avance
11    // conjunto de instrucciones
12    //}
13
14    //Ejercicio 1
15    cout << "Impresión de números del 1 al 10" << endl;
16    cout << "Forma ascendente" << endl;
17    for (int i = 1; i <= 10; i++){
18        cout << i << endl;
19    }
20    cout << "\n";
21
22    cout << "Forma descendente" << endl;
23    for (int j = 10; j >= 1; j--){
24        cout << j << endl;
25    }
26
27    return 0;
28 }
29
```

The output window shows the following text:

```
D:\Documentos\2do Semestre\programaci4n\laboratorios\Laboratorio...
Impresión de números del 1 al 10
Forma ascendente
1
2
3
4
5
6
7
8
9
10
Forma descendente
10
9
8
7
6
5
4
3
2
1
-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicio 2:



The screenshot shows a C++ IDE with several tabs. The active tab is 'Operadores_Guerra_Josué.cpp', which contains the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     // For
7     //Estructura
8     // for (exp1; expresion logica; exp2){
9     //exp1- inicializador el interador
10    //exp2- iterador avance
11    // conjunto de instrucciones
12    //}
13
14    // Ejercicio 2: Usar for para imprimir las potencias de 3 y 5 de forma inversa
15
16    cout << "Potencias en forma inversa" << endl;
17    cout << "Potencias de 3 y 5" << endl;
18    cout << "Ingrese la cantidad de potencias que desea imprimir: " << endl;
19    int potencias;
20    cin >> potencias;
21
22    cout << "Potencias de 3 en orden inverso: " << endl;
23    for (int exponente = potencias - 1; exponente >= 0; --exponente)
24    {
25        int resultado = 1;
26        for (int i = 0; i < exponente; ++i)
27        {
28            resultado *= 3;
29        }
30        cout << resultado << " ";
31    }
32    cout << "\n\nPotencias de 5 en orden inverso: " << endl;
33    for (int exponente = potencias - 1; exponente >= 0; --exponente)
34    {
35        int resultado = 1;
36        for (int i = 0; i < exponente; ++i)
37        {
38            resultado *= 5;
39        }
40        cout << resultado << " ";
41    }
42
43    return 0;
44 }
```

The output window shows the following text:

```
D:\Documentos\2do Semestre\programaci4n\laboratorios\Laboratorio6 20-12-...
Potencias en forma inversa
Potencias de 3 y 5
Ingrese la cantidad de potencias que desea imprimir:
8
Potencias de 3 en orden inverso:
2187 729 243 81 27 9 3 1
Potencias de 5 en orden inverso:
78125 15625 3125 625 125 25 5 1
-----
Process exited with return value 0
Press any key to continue . . .
```

• BUCLES ANIDADOS

Ejercicio 1 y 2:

```

1 #include <iostream>
2 #include <windows.h>
3 #include <stdlib.h>
4 using namespace std;
5
6 int main ()
7 {
8     //Ejercicio 1 de ejemplo
9     cout<<"Ejercicio 1: "<<endl;
10    for (int i = 1; i<= 4; i++){
11        for (int j = 1; j<= 3; j++){
12            cout<<" ";
13        }
14    }
15
16    //Ejercicio 2 tabla de multiplicar
17    // 1*1=1
18    // 1*2=2
19    // 1*3=3
20    cout<<"\n";
21    cout<<"Ejercicio 2"<<endl;
22    int i, j; //Declarar dentro dle for es una buena pr?ctica
23
24    //Controla el numero de veces que se repite
25    for(i = 1; i<=9; i++){
26        cout<<"tabla de multiplicar del "<<i<<endl;
27        for(j = 1; j<=10; j++){
28            cout<<i<<" x "<<j<<" = "<<i*j<<endl;
29        }
30        cout<<endl<<endl;
31        //sirve para poder pausar el sistema
32        //y que solo con una tecla se pueda continuar
33        system (" pause ");
34
35        //Limpia la pantalla
36        system (" cls ");
37    }
38
39    return 0;
40 }

```

Ejercicio 1:

```

* * * * *
Ejercicio 2
tabla de multiplicar del 1
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
Presione una tecla para continuar . . .

```

tabla de multiplicar del 2

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
Presione una tecla para continuar . . .

```

tabla de multiplicar del 6

```

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
Presione una tecla para continuar . . .

```

tabla de multiplicar del 9

```

9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
Presione una tecla para continuar . . .

```

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:

```

1 #include <iostream>
2 #include <windows.h>
3 #include <stdlib.h>
4 using namespace std;
5
6 int main ()
7 {
8     //Ejercicio 3 Cronometro
9     //Contorlar el tiempo
10    //Hacer que se duerma durante un determinado periodo
11    //de tiempo
12    //timepo en milisegundos
13
14    int h,min,seg,x;
15    x =1000;
16    for(h=0; h<24; h++){
17        for(min=0; min<60; min++){
18            for(seg=0; seg<60; seg++){
19                cout<<h<<" : "<<min<<" : "<<seg<<"\r" ;
20                //Duerma un segundo cada
21                Sleep(x);
22            }
23        }
24    }
25    system ("pause");
26
27    //lr sirve para qeu se remplace encima,
28
29    return 0;
30 }

```

0 : 0 : 12

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

Ejercicio 4, 5, 6:

```

Bucles_anidados_Ejercicio1_Guerra_Josué.cpp Bucles_anidados_Ejercicio3_Guerra_Josué.cpp Bucles_anidados_Ejercicio4_3_Guerra_Josué.cpp
1 #include <iostream>
2 #include <windows.h>
3 #include <stdlib.h>
4
5 using namespace std;
6
7 int main ()
8 {
9     cout<<"Ejercicio 4, 5, 6"<<endl;
10    cout<<" ";
11    //Ejercicio 4 FILAS Y Columnas de un rectangulo
12    cout<<" Filas y columnas de un rectangulo"<<endl;
13    int a, b;
14    int c, d;
15    cout<<"inserte el numero de filas "<<endl;
16    cin>>a;
17    cout<<"inserte el numero de columnas "<<endl;
18    cin>>b;
19
20    for (int c = 0; c < a ; c ++){
21        cout<<endl;
22        for (int d = 0; d < b; d++){
23            cout<<" * ";
24        }
25    }
26
27    //Ejercicio 5 Filas y columnas de rectangulo
28    //Solo Imprimira el borde del rectangulo
29    cout<<"\n";
30    cout<<" ";
31    cout<<"Impresión de rectangulo solo el borde"<<endl;
32    int m, j;
33    int f, l;
34
35    cout<<"inserte el numero de filas "<<endl;
36    cin>>m;
37    cout<<"inserte el numero de columnas "<<endl;
38    cin>>j;
39
40    for (int f = 0 ; f < m ; f++)
41    {
42        for (int l = 0; l < j ; l++)
43        {
44            if (f == 0 || f == m - 1 || l == 0 || l == j - 1 )
45            {
46                cout<<"*";
47            }
48            else
49            {
50                cout<<" ";
51            }
52        }
53        cout<<endl;
54    }
55    //Ejercicio 6 Impresión solo del centro de un cuadrado
56    //Solo se imprime el centro del caudarado
57    //Deben ser sus lados iguales
58    cout<<"Impresión del centro un cuadrado"<<endl;
59    int filas, columnas;
60
61    do
62    {
63        cout<<"inserte el numero de filas "<<endl;
64        cin>>filas;
65        cout<<"inserte el numero de columnas "<<endl;
66        cin>>columnas;
67        if (filas != columnas)
68        {
69            cout << "Los números deben ser iguales. Intente nuevamente." << endl;
70        }
71    } while (filas != columnas);
72
73    for (int i = 0; i < filas; ++i)
74    {
75        for (int j = 0; j < columnas; ++j)
76        {
77            if (i == j || i == filas - 1 - j)
78            {
79                cout << "*";
80            }
81            else
82            {
83                cout << " ";
84            }
85        }
86        cout << endl;
87    }
88
89    return 0;
90 }
91

```

```

Ejercicio 4, 5, 6
Filas y columnas de un rectangulo
inserte el numero de filas
5
inserte el numero de columnas
8

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Impresión de rectangulo solo el borde
inserte el numero de filas
6
inserte el numero de columnas
5

*****
*   *
*   *
*   *
*   *
*   *
*****

Impresión del centro un cuadrado
inserte el numero de filas
8
inserte el numero de columnas
8

*       *
*      *
*     *
*    *
*   *
*  *
* *
*
*
*
*

-----
Process exited with return value 0
Press any key to continue . . .

```

```

Filas y columnas de un rectangulo
inserte el numero de filas
5
inserte el numero de columnas
8

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Impresión de rectangulo solo el borde
inserte el numero de filas
6
inserte el numero de columnas
5

*****
*
*
*
*
*****

Impresión del centro un cuadrado
inserte el numero de filas
8
inserte el numero de columnas
8

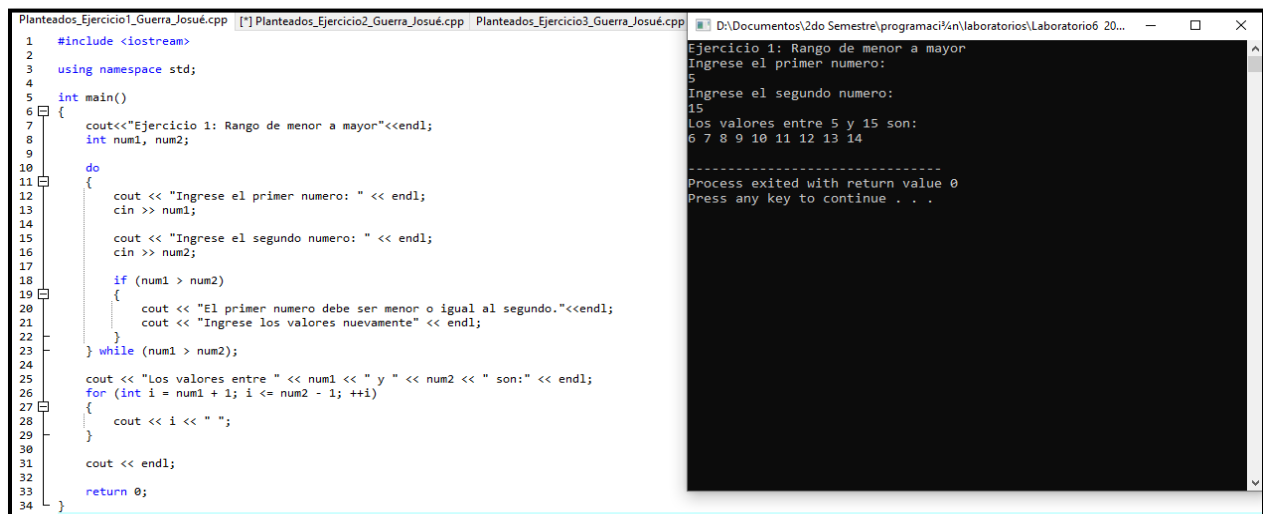
* * * * *
* * * *
* * *
* *
*
* *
* *
* *

```

EJERCICIOS PLANTEADOS

Ejercicios planteados por la ingeniera

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



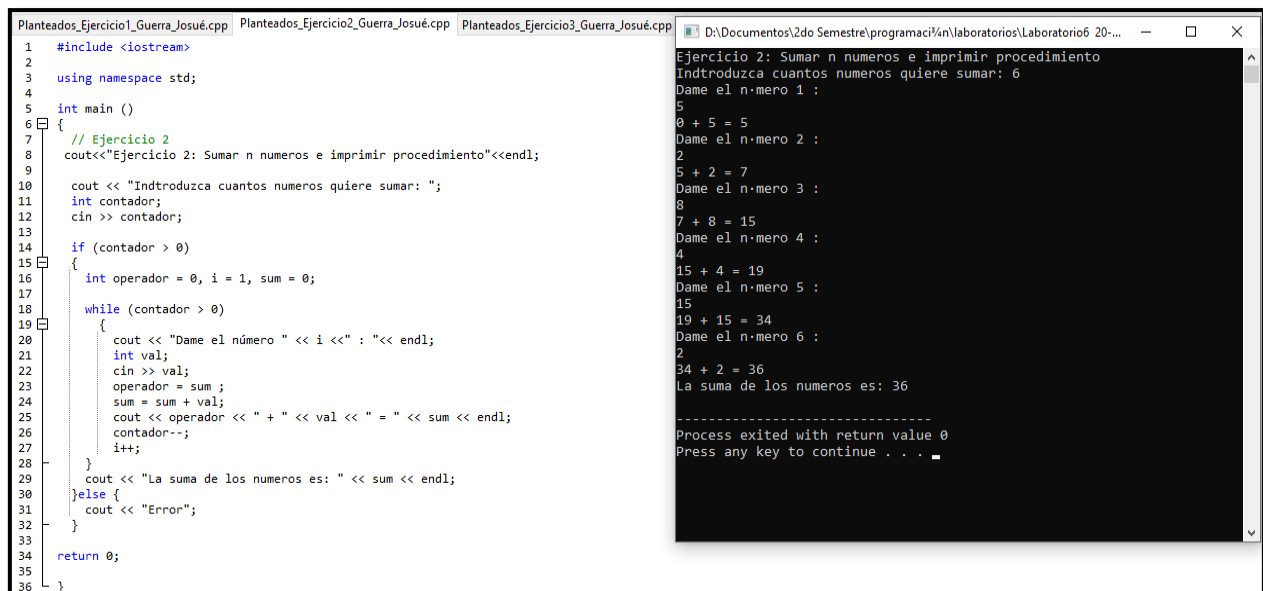
```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout<<"Ejercicio 1: Rango de menor a mayor"<<endl;
8     int num1, num2;
9
10    do
11    {
12        cout << "Ingrese el primer numero: " << endl;
13        cin >> num1;
14
15        cout << "Ingrese el segundo numero: " << endl;
16        cin >> num2;
17
18        if (num1 > num2)
19        {
20            cout << "El primer numero debe ser menor o igual al segundo."<<endl;
21            cout << "Ingrese los valores nuevamente" << endl;
22        }
23    } while (num1 > num2);
24
25    cout << "Los valores entre " << num1 << " y " << num2 << " son:" << endl;
26    for (int i = num1 + 1; i <= num2 - 1; ++i)
27    {
28        cout << i << " ";
29    }
30
31    cout << endl;
32
33    return 0;
34 }
```

Ejercicio 1: Rango de menor a mayor
Ingrese el primer numero:
5
Ingrese el segundo numero:
15
Los valores entre 5 y 15 son:
6 7 8 9 10 11 12 13 14

Process exited with return value 0
Press any key to continue . . .

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

- Sume N números ingresados por el usuario.



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main ()
6 {
7     // Ejercicio 2
8     cout<<"Ejercicio 2: Sumar n numeros e imprimir procedimiento"<<endl;
9
10    cout << "Introduzca cuantos numeros quiere sumar: ";
11    int contador;
12    cin >> contador;
13
14    if (contador > 0)
15    {
16        int operador = 0, i = 1, sum = 0;
17
18        while (contador > 0)
19        {
20            cout << "Dame el número " << i << " : "<< endl;
21            int val;
22            cin >> val;
23            operador = sum ;
24            sum = sum + val;
25            cout << operador << " + " << val << " = " << sum << endl;
26            contador--;
27            i++;
28        }
29        cout << "La suma de los numeros es: " << sum << endl;
30    }else {
31        cout << "Error";
32    }
33
34    return 0;
35 }
36 }
```

Ejercicio 2: Sumar n numeros e imprimir procedimiento
Indroduzca cuantos numeros quiere sumar: 6
Dame el n-mero 1 :
5
0 + 5 = 5
Dame el n-mero 2 :
2
5 + 2 = 7
Dame el n-mero 3 :
8
7 + 8 = 15
Dame el n-mero 4 :
4
15 + 4 = 19
Dame el n-mero 5 :
15
19 + 15 = 34
Dame el n-mero 6 :
2
34 + 2 = 36
La suma de los numeros es: 36

Process exited with return value 0
Press any key to continue . . .

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.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main ()
6 {
7     // Ejercicio 3: Promedio de la suma de n numeros
8     // Pedir al usuario n números para realizar la suma
9     // Después se hace el calculo del promedio
10    cout<<"Ejercicio 3: Promedio de la suma de numeros"<<endl;
11    do
12    {
13        cout << "Introduzca cuantos numeros quiere sumar: ";
14        int contador;
15        cin >> contador;
16
17        if (contador <=0)
18        {
19            cout << "El numero a sumar debe ser mayor a 0"<<endl;
20            cout << "Inserte nuevamente la cantidad : ";
21            cin>>contador;
22        }
23
24        float promedio, cont = contador, operador = 0;
25        int num = 1, suma;
26
27        while (contador > 0)
28        {
29            cout << "Dame el numero "<<num<< " : ";
30            cin >> suma;
31            operador = operador + suma;
32            contador--;
33            num++;
34        }
35
36        cout << "La suma de los numeros es: " << operador << endl;
37        promedio = operador / cont;
38        cout<<operador<< " / " <<cont<< " = "<< promedio<<endl;
39        cout <<"El promedio de la suma es: "<<promedio;
40
41    }while (false);
42    return 0;
43 }
44

```

```

Ejercicio 3: Promedio de la suma de numeros
Introduzca cuantos numeros quiere sumar: 5
Dame el numero 1 :12
Dame el numero 2 :10
Dame el numero 3 :6
Dame el numero 4 :8
Dame el numero 5 :5
La suma de los numeros es: 41
41 / 5 = 8.2
El promedio de la suma es: 8.2
-----
Process exited with return value 0
Press any key to continue . . .

```

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.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int num;
8     cout<<" Ejercicio 4: Calculo de factorial"<<endl;
9     cout<< "Ingrese un número para calcular su factorial: "<<endl;
10    cin >> num;
11
12    if (num < 0)
13    {
14        cout << "El factorial no puede ser negativo." << endl;
15    }
16    else
17    {
18        int factorial = 1;
19
20        cout << "Fórmula del factorial: " << num << "! = ";
21
22        for (int i = 1; i <=num; ++i)
23        {
24            cout<<i;
25            if (i < num)
26            {
27                cout<< " * ";
28            }
29        }
30        cout <<endl;
31
32        for (int i = 1; i <= num; ++i)
33        {
34            factorial *= i;
35        }
36        cout << "El factorial de " << num << " es: " << factorial << endl;
37
38    }
39    return 0;
40 }

```

```

Ejercicio 4: Calculo de factorial
Ingrese un n-mero para calcular su factorial:
7
Fórmula del factorial: 7! = 1 * 2 * 3 * 4 * 5 * 6 * 7
El factorial de 7 es: 5040
-----
Process exited with return value 0
Press any key to continue . . .

```

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.

- Calcula el valor de: $2^1 + 2^2 + 2^3 + \dots + 2^n$

```

1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main()
7 {
8     // Ejercicio 5: Suma de las n potencias de 2
9     // Pedir al usuario n números para realizar la suma
10    // Se realiza el calculo de las n potencias
11
12    cout<<"Ejercicio 5: Calculo de la n potencia"<<endl;
13    int n;
14
15    cout<<"Potencias de 2^1 + 2^2 + ... + 2^n:"<<endl;
16    cout<<"Ingrese un número para calcular la n potencia de 2"<<endl;
17    cin >> n;
18
19    if (n < 1)
20    {
21        cout << "El número debe ser mayor o igual a 1." << endl;
22    }
23    else
24    {
25        int resultado = 0;
26        cout<<"Su formula es: "<<endl;
27
28        for (int i = 1; i <= n; ++i)
29        {
30            cout<<" 2^"<<i;
31            if (i<n)
32            {
33                cout<<" + ";
34            }
35
36            resultado += pow(2, i);
37        }
38        cout<<endl;
39        cout << "El resultado de la suma es: " << resultado << endl;
40    }
41
42    return 0;
43 }

```

```

Ejercicio 5: Calculo de la n potencia
Potencias de 2^1 + 2^2 + ... + 2^n:
Ingrese un n-mero para calcular la n potencia de 2
9
Su formula es:
2^1+ 2^2+ 2^3+ 2^4+ 2^5+ 2^6+ 2^7+ 2^8+ 2^9
El resultado de la suma es: 1022

-----
Process exited with return value 0
Press any key to continue . . .

```

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: $1 - 2 + 3 - 4 + 5 - 6 \dots n$.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // Ejercicio 6: Suma de la serie
8     // Pedir al usuario n números para realizar la suma
9     // Calcula el resultado de: 1-2+3-4+5-6...n
10
11    cout<<"Ejercicio 6: Suma de la serie"<<endl;
12    int n;
13
14    do{
15        cout<< "Serie de enteros 1 -2 +3 -4 +5... +n -n: "<<endl;
16        cout<< "Ingrese hasta que numero desea calcular: "<<endl;
17        cin >> n;
18
19        if (n < 1)
20        {
21            cout << "El número debe ser mayor o igual a 1." << endl;
22        }
23    }while (n<1);
24
25    int resultado = 0;
26    int i = 1;
27    int signo = 1;
28    cout << "Fórmula de la serie: "<<endl;
29
30    do
31    {
32        cout << (signo == 1 ? "+" : "-") << i;
33        if (i < n)
34        {
35            cout << (signo == 1 ? " " : " + ");
36        }
37        resultado += signo * i;
38        signo *= -1;
39        i++;
40    } while (i <= n);
41
42    cout << endl;
43    cout << "El resultado de la serie es: " << resultado << endl;
44
45    return 0;
46 }

```

```

Ejercicio 6: Suma de la serie
Serie de enteros 1 -2 +3 -4 +5... +n -n:
Ingrese hasta que numero desea calcular:
16
Fórmula de la serie:
1 -2 +3 -4 +5 -6 +7 -8 +9 -10 +11 -12 +13 -14 +15 -16
El resultado de la serie es: -8

-----
Process exited with return value 0
Press any key to continue . . .

```

La serie planteada muestra la suma y resta de todos los números naturales, entonces en el programa se 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.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // Ejercicio 7: Serie de fibonacci
8     // Pedir al usuario n números
9     // Se mostrara hasta ese número o el anterior si es el caso
10    cout << "Ejercicio 7: Serie de fibonacci" << endl;
11
12    int n;
13    cout << "\n";
14
15    do
16    {
17        cout << " Para mostrar la serie de Fibonacci hasta n valor " << endl;
18        cout << " Ingrese un número: " << endl;
19        cin >> n;
20
21        if (n < 0)
22        {
23            cout << "Ingrese el número correcto" << endl;
24        }
25    } while (n < 0);
26
27    int a = 0, b = 1, c;
28
29    cout << "Serie de Fibonacci hasta " << n << ": ";
30
31    while (a <= n)
32    {
33        cout << a << " ";
34        c = a + b;
35        a = b;
36        b = c;
37    }
38
39    cout << endl;
40
41    return 0;
42 }
43

```

Ejercicio 7: Serie de fibonacci

Para mostrar la serie de Fibonacci hasta n valor
Ingrese un n-mero:
21
Serie de Fibonacci hasta 21: 0 1 1 2 3 5 8 13 21

Process exited with return value 0
Press any key to continue . . .

- 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.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int numero, numero2, suma;
7     cout << "Ingrese su nombre: " << endl;
8     string nombre;
9     cin >> nombre;
10
11     for (int j = 1; j <= 10; j++) {
12         suma = 0;
13         cout << "Ingrese el número " << j << ": " << endl;
14         cout << "Al ingresar 0 el programa terminara " << endl;
15         cin >> numero2;
16
17         if (numero2 == 0) {
18             cout << nombre << endl;
19             cout << "Vuelva Pronto" << endl;
20             cout << "Gracias por usar" << endl;
21             cout << "nuestro Sistema" << endl;
22             cout << "Creado Por Josue Guerra";
23             break;
24         }
25
26         for (int i = 1; i <= numero2; i++) {
27             if (numero2 % i == 0) {
28                 suma += i;
29                 cout << i << " + ";
30             }
31         }
32
33         cout << "\b\b = " << numero2 << endl;
34
35         if (suma == numero2) {
36             cout << "El numero " << numero2 << " es un numero perfecto" << endl;
37         } else {
38             cout << "El numero " << numero2 << " no es un numero perfecto" << endl;
39         }
40     }
41
42    return 0;
43 }
44

```

Ingrese su nombre:
J0sue

Ingrese el n-mero 1:
Al ingresar 0 el programa terminara
6
 $1 + 2 + 3 = 6$
El numero 6 es un numero perfecto

Ingrese el n-mero 2:
Al ingresar 0 el programa terminara
496
 $1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248 = 496$
El numero 496 es un numero perfecto

Ingrese el n-mero 3:
Al ingresar 0 el programa terminara
15
 $1 + 3 + 5 = 9$
El numero 15 no es un numero perfecto

Ingrese el n-mero 4:
Al ingresar 0 el programa terminara
0
J0sue

Vuelva Pronto

Gracias por usar

nuestro Sistema

Creado Por Josue Guerra

Process exited with return value 0
Press any key to continue . . .

- ```
Ejemplos_Ejercicio1_Guerra_Josué.cpp Ejemplos_Ejercicio101_1_Guerra_Josué.cpp Ejemplos_Ejercicio2_Guerra_Josué.cpp [*] Ejemplos_Ejercicio3_Gue...
Ejemplos_Ejercicio3.2_Guerra_Josué.cpp Untitled3.cpp Planteados_Ejercicio9_Guerra_Josué.cpp

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7 // Ejercicio 9: Arbolito de navidad
8 //Pedir al usuario n número
9 //Se mostrara el arbolito segun el número
10 cout<<"Ejercicio 9: Arbolito navideño"<<endl;
11 cout << "Ingrese el número de ramas del Arbolito de Navidad: ";
12 int numRamas;
13 cin >> numRamas;
14 if (numRamas <= 0)
15 {
16 cout << "El número debe ser diferente de 0." << endl;
17 }
18 for (int rama = 1; rama <= numRamas; ++rama)
19 {
20 for (int espacio = 0; espacio < numRamas - rama; ++espacio)
21 {
22 cout << " ";
23 }
24 for (int asterisco = 0; asterisco < 2 * rama - 1; ++asterisco)
25 {
26 cout << "**";
27 }
28 cout << endl;
29 }
30
31
32 int espacioTronco = numRamas - 1;
33 for (int tronco = 0; tronco < numRamas - 1; ++tronco)
34 {
35 cout << " ";
36 }
37 cout << "| " << endl;
38
39 return 0;
40 }
```

- 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 ();**
- putchar (): Imprimir información en la ventana de salida, leyendo solo un carácter a la vez. /**int putchar (int c);**
- cgets (): leer cadena de caracteres hasta que se encuentre el retorno CR y salto de línea LF, reemplazando 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 textcolor(int color);`
- `textbackground ()`: Sirve para cambiar el color de fondo. / `void textbackground(<color>);`

1. Figura 1: uso de las 5 primeras funciones `clrscr()`, `getch()`, `getche()`, `putch()`, `cgets()`

The screenshot shows a C++ program named `Ejemplos_Ejercicio1_Guerra_Josué.cpp` and its output in a terminal window. The program includes `<stdio.h>` and `<conio.h>`. It defines `clrscr()` using `system("cls")`, `putch()` using `putchar()`, and `cgets()` using `fgets()`. In the `main()` function, it clears the screen, prints a message, reads a string with `cgets()`, prints it, and waits for a key press with `getch()`.

```

1 #include <stdio.h>
2 #include <conio.h>
3
4 void clrscr() {
5 int _cdecl system(const char * cls);
6 }
7
8 void putch(char ch) {
9 putchar(ch);
10 }
11
12 char* cgets(char *str, int size) {
13 return fgets(str, size, stdin);
14 }
15
16 int main() {
17 clrscr();
18
19 putch('H');
20 putch('O');
21 putch('L');
22 putch('A');
23 putch('X');
24 putch('D');
25 putch(',');
26
27 printf(" Que la fuerza te acompañe!\n");
28
29 char input[50];
30 printf("Ingresa un texto: ");
31 cgets(input, sizeof(input)); // Utiliza fgets en lugar de gets
32
33 printf("Tu ingresaste: %s\n", input);
34
35 getch(); // Espera una tecla
36
37 return 0;
38 }

```

Terminal output:

```

D:\Documentos\2do Semestre\programaci3n\labo...
HOLAXD, Que la fuerza te acompa!
Ingresa un texto: a
Tu ingresaste: a

Process exited with return value 0
Press any key to continue . . .

```

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

The screenshot shows a C++ program named `Ejemplos_Ejercicio1_Guerra_Josué.cpp` and its output in a terminal window. The program includes `<windows.h>`, `<conio.h>`, `<stdio.h>`, and `<stdarg.h>`. It defines custom versions of `clrscr()`, `cscanf()`, and `cprintf()` using `system()`, `va_list`, `vsprintf()`, `textcolor()`, and `textbackground()`. In the `main()` function, it uses these custom functions to clear the screen, read an integer with `custom_cscanf()`, print a message with `custom_cprintf()`, and wait for a key press with `getch()`.

```

1 #include <windows.h>
2 #include <conio.h>
3 #include <stdio.h>
4 #include <stdarg.h>
5
6 void custom_clrscr() {
7 int _cdecl system(const char * cls);
8 }
9
10 void custom_cscanf(const char *format, ...) {
11 va_list args;
12 va_start(args, format);
13 vscanf(format, args);
14 va_end(args);
15 }
16
17 void custom_cprintf(int text_color, int background_color, const char *format, ...) {
18 void textcolor(int YELLOW);
19 void textbackground(int BLUE);
20
21 va_list args;
22 va_start(args, format);
23 vsprintf(format, args);
24 va_end(args);
25 }
26
27
28 int main() {
29 custom_clrscr();
30
31 printf("INGRESA UN NUMERO: ");
32
33 int num;
34 custom_cscanf("%d", &num);
35
36 custom_cprintf(YELLOW, BLUE, "Has ingresado el: %d\n", num);
37
38 printf("Presiona una tecla para salir...");
39
40 getch(); // Esperar a que se presione una tecla
41
42 return 0;
43 }

```

Terminal output:

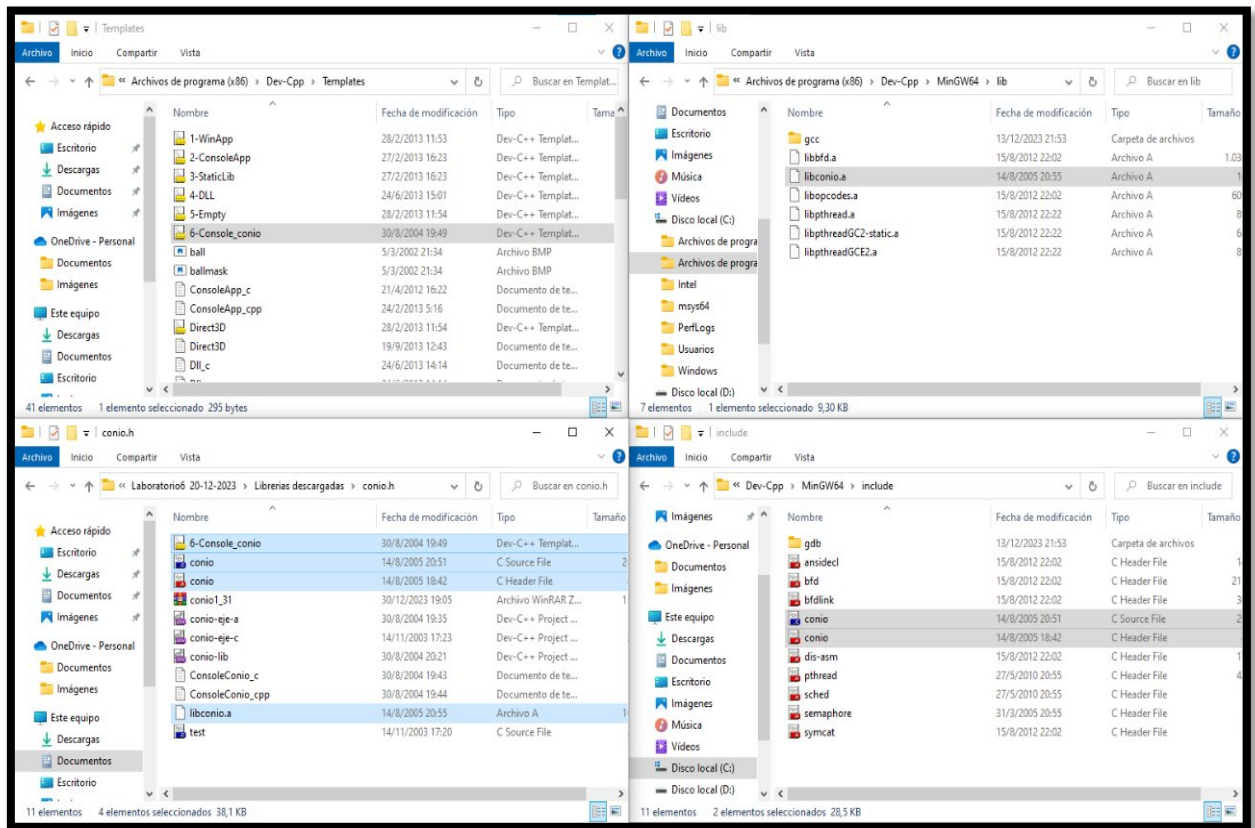
```

D:\Documentos\2do Semestre\programaci3n\labo...
INGRESA UN NUMERO: 3
Has ingresado el: 3
Presiona una tecla para salir...

Process exited with return value 0
Press any key to continue . . .

```

Figura 3: Instalación de las librerías en DEV C++, 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:

```

Ejemplos_Ejercicio1_Guerra_Josué.cpp Ejemplos_Ejercicio1.1_Guerra_Josué.cpp Ejemplos_Ejercicio2_Guerra_Josué.cpp
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7 int intValue = 12345;
8 double doubleValue = 1234.56789;
9
10 //Se usa setw y setfill para formatear la salida
11 cout << setw(10) << setfill('0') << intValue << endl;
12
13 // Uso de setprecision y fixed para formatear la salida de un número de punto flotante
14 cout << fixed << setprecision(2) << doubleValue << endl;
15
16 return 0;
17 }

```

Output window showing the results of the program execution:

```

D:\Documentos\2do Semestre\programaci3n\labor...
0000012345
1234.57

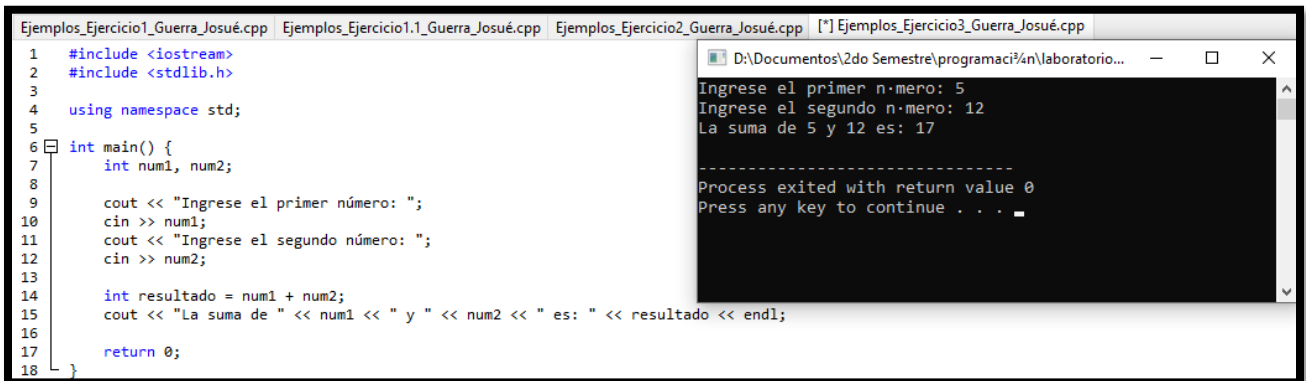
Process exited with return value 0
Press any key to continue . . .

```

### ○ 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



```
1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 int main() {
7 int num1, num2;
8
9 cout << "Ingrese el primer número: ";
10 cin >> num1;
11 cout << "Ingrese el segundo número: ";
12 cin >> num2;
13
14 int resultado = num1 + num2;
15 cout << "La suma de " << num1 << " y " << num2 << " es: " << resultado << endl;
16
17 return 0;
18 }
```

Output:

```
Ingrese el primer n-mero: 5
Ingrese el segundo n-mero: 12
La suma de 5 y 12 es: 17

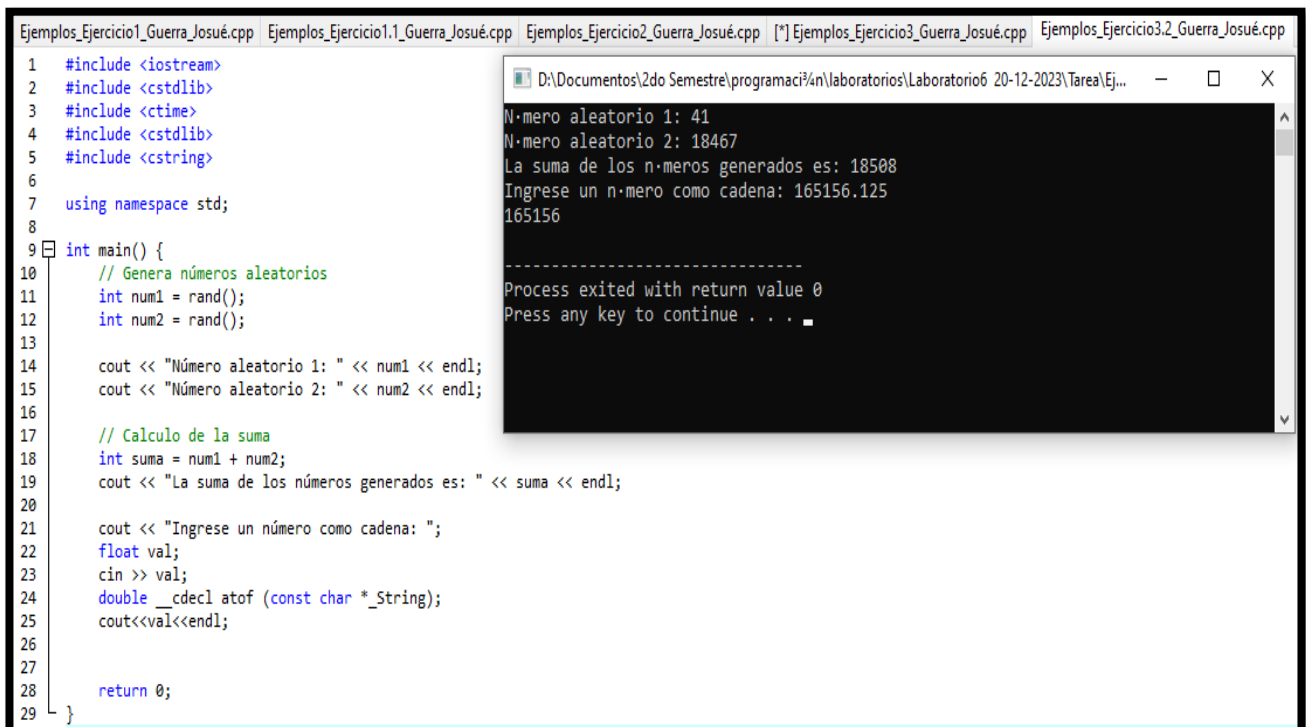
Process exited with return value 0
Press any key to continue . . .
```

Figura 3.2:

Se usan funciones como lo son:

Rand: para creación de números aleatorios.

Atof: para convertir una cadena en doble.



```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 #include <string>
5 #include <cstring>
6
7 using namespace std;
8
9 int main() {
10 // Genera números aleatorios
11 int num1 = rand();
12 int num2 = rand();
13
14 cout << "Número aleatorio 1: " << num1 << endl;
15 cout << "Número aleatorio 2: " << num2 << endl;
16
17 // Calculo de la suma
18 int suma = num1 + num2;
19 cout << "La suma de los números generados es: " << suma << endl;
20
21 cout << "Ingrese un número como cadena: ";
22 float val;
23 cin >> val;
24 double __cdecl atof (const char *_String);
25 cout<<val<<endl;
26
27
28 return 0;
29 }
```

Output:

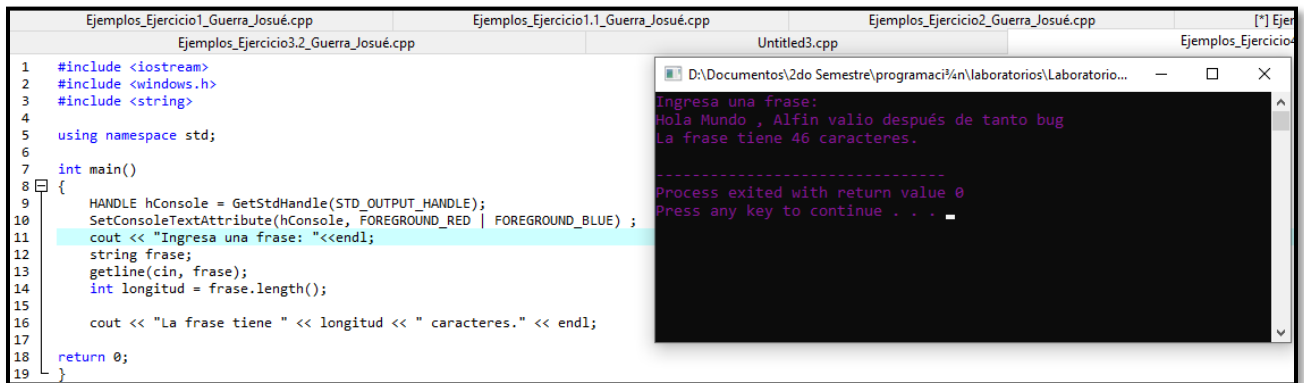
```
N-mero aleatorio 1: 41
N-mero aleatorio 2: 18467
La suma de los n-meros generados es: 18508
Ingrese un n-mero como cadena: 165156.125
165156

Process exited with return value 0
Press any key to continue . . .
```



## ○ 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].



```
1 #include <iostream>
2 #include <windows.h>
3 #include <string>
4
5 using namespace std;
6
7 int main()
8 {
9 HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
10 SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_BLUE);
11 cout << "Ingresa una frase: "<<endl;
12 string frase;
13 getline(cin, frase);
14 int longitud = frase.length();
15
16 cout << "La frase tiene " << longitud << " caracteres." << endl;
17
18 return 0;
19 }
```

## 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:  
<https://github.com/JosueGuerra2023B/programacion2023B/tree/main/Laboratorio6%20%2020-12-2023>
  - Enlace OneDrive:  
[Laboratorio6 20-12-2023](#)

## Bibliografía

- [1] Freedigitales, «Programa videojuegos,» 2023. [En línea]. Available: <https://programavideojuegos.blogspot.com/2013/05/la-libreria-conioh.html>. [Último acceso: 28 12 2023].
- [2] Dbush, «Stack overflow,» 19 01 2020. [En línea]. Available: <https://stackoverflow.com/questions/59812014/why-use-conio-h>. [Último acceso: 28 12 2023].
- [3] Ballena, «BALLENA,» 2023. [En línea]. Available: [https://techsupportwhale.com/conio-h-c-plus-plus-library/#google\\_vignette](https://techsupportwhale.com/conio-h-c-plus-plus-library/#google_vignette). [Último acceso: 20 12 2023].
- [4] Ballena, «Soporte técnico ballena,» 06 10 2023. [En línea]. Available: <https://techsupportwhale.com/conio-h-c-plus-plus-library/#cgets>. [Último acceso: 20 12 2023].
- [5] C. Rolon, «Youtube,» 30 11 2014. [En línea]. Available: <https://youtu.be/S4az7cVyvT0?si=oJ60Ia51tK6Y3xzE>. [Último acceso: 28 12 2023].
- [6] Microsoft, «Microsoft,» [En línea]. Available: <https://learn.microsoft.com/en-us/cpp/standard-library/iomanip?view=msvc-170>. [Último acceso: 28 12 2023].
- [7] Arduinoque, «Arquinoque,» 2023. [En línea]. Available: <https://arduinoque.com/arduino/para-que-sirve-la-libreria-stdlib-h/>. [Último acceso: 28 12 2023].
- [8] cplusplus, «cplusplus.com,» 16 08 2019. [En línea]. Available: <https://cplusplus.com/reference/cstdlib/>. [Último acceso: 28 12 2023].
- [9] B. Blackened, «StackOverflow,» 15 07 2019. [En línea]. Available: <https://es.stackoverflow.com/questions/278904/diferencia-entre-windows-h-stdafx-h>. [Último acceso: 28 12 2023].