



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



PROGRAMACIÓN

ASIGNATURA:

PROGRAMACIÓN

PROFESOR:

Ing. Yadira Franco Rocha

PERÍODO ACADÉMICO:

2025-B

TÍTULO

EJERCICIOS REPASO



ESTUDIANTES:

➤ Lozano Yausen Karen Brigitte

Introducción

El objetivo de este taller fue reforzar los conceptos de programación en C++ mediante el desarrollo de ejercicios prácticos que involucren **arreglos, vectores, matrices, estructuras, funciones, paso por referencia y archivo**.

El taller buscó que el estudiante adquiriera habilidades para:


- Gestionar datos en estructuras lineales y bidimensionales.
- Aplicar condiciones y ciclos para cálculos específicos.
- Manejar archivos para almacenar y recuperar información.
- Implementar funciones para modularizar el código y evitar redundancia.

Desarrollo

En este taller se realizaron ejercicios que abarcan distintas temáticas:

Ejercicio 1

Elabore un programa en C++ que solicite un número entero e indique si el número pertenece al rango de 10 a 50, mostrando un mensaje cuando el valor sea válido y otro cuando no lo sea.



The screenshot shows a C++ program in a code editor and its execution in a terminal window. The code is as follows:

```
1 //Ejercicio 1
2 //Elabore un programa en C++ que solicite un número entero e
3 //indique si el número pertenece al rango de 10 a 50,
4 //mostrando un mensaje cuando el valor sea válido y
5 //otro cuando no lo sea.
6 #include <iostream>
7 using namespace std;
8
9
10 int main(){
11     int numero;
12     cout<<"Ingrese un número entero:";
13     cin>>numero;
14     if(numero>=10 && numero<=50){
15         cout<<"El número pertenece al rango de 10 a 50"<<endl;
16     }
17     else{
18         cout<<"El número no pertenece al rango"<<endl;
19     }
20     return 0;
21 }
```

The terminal window shows the following output:

```
C:\Users\USUARIO\OneDrive\ x + - □ x
Ingrese un número entero:45
El número pertenece al rango de 10 a 50
-----
Process exited after 3.135 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 2

Desarrolle un programa que solicite un número entero y determine si el número NO se encuentra en el rango de 10 a 50, mostrando un mensaje de advertencia cuando esté fuera del rango permitido.

```

1 //Ejercicio 2
2 //Desarrolle un programa que solicite un número entero
3 //y determine si el número NO se encuentra en el rango
4 //de 10 a 50, mostrando un mensaje de advertencia cuando
5 //esté fuera del rango permitido.
6
7 #include <iostream>
8
9 using namespace std;
10 int main(){
11     int numero;
12     cout<<"Ingrese un numero entero:";
13     cin>>numero;
14
15     if(numero<10 || numero>50){
16         cout<<"El numero no se encuentra en el rango de 10 a 50."<<endl;
17     }
18     else{
19         cout<<"El numero pertenece al rango de 10 a 50."<<endl;
20     }
21     return 0;
22 }

```

```

C:\Users\USUARIO\OneDrive\
Ingrese un numero entero:5
El numero no se encuentra en el rango de 10 a 50.

-----
Process exited after 2.889 seconds with return value 0
Presione una tecla para continuar . . . |

```

B. EJERCICIOS SOLO CON FOR

Ejercicio 3

Desarrolle un programa que muestre en pantalla los números del 1 al 10, pero indicando al final cuántos números fueron mostrados, utilizando únicamente la estructura for.

```

1 //Ejercicio 3
2 //Desarrolle un programa que muestre en pantalla los números
3 //del 1 al 10, pero indicando al final cuántos números fueron
4 //mostrados, utilizando únicamente la estructura for.
5
6 #include <iostream>
7
8 using namespace std;
9
10 int main (){
11     int contador=0;
12     cout<<"NUMEROS DEL 1 AL 10"<<endl;
13     for(int i=1; i<=10; i++){
14         cout<<i<<endl;
15         contador++;
16     }
17     cout<<"LA CANTIDAD DE NUMEROS INGRESADOS SON:"<<contador<<endl;
18     return 0;
19 }

```

```

C:\Users\USUARIO\OneDrive\
NUMEROS DEL 1 AL 10
1
2
3
4
5
6
7
8
9
10
LA CANTIDAD DE NUMEROS INGRESADOS SON:10

-----
Process exited after 0.7699 seconds with return value 0
Presione una tecla para continuar . . . |

```

Ejercicio 4

Realice un programa que muestre los números pares del 2 al 20 utilizando un ciclo for y muestre también la suma de dichos números.

```

1 //Ejercicio 4
2 //Realice un programa que muestre los números pares del 2 al 20
3 //utilizando un ciclo for y muestre también la suma de dichos números.
4
5 #include <iostream>
6
7 using namespace std;
8
9 int main (){
10     int suma=0;
11     cout<<"NUMEROS PARES 2-20"<<endl;
12     for(int i=2; i<=20; i+=2){
13         cout<<i<<endl;
14         suma+=i;
15     }
16     cout<<"La suma de estos numeros es:"<<suma;
17     return 0;
18 }

```

```

C:\Users\USUARIO\OneDrive\
NUMEROS PARES 2-20)
2
4
6
8
10
12
14
16
18
20
La suma de estos numeros es:110

-----
Process exited after 0.7886 seconds with return value 0
Presione una tecla para continuar . . . |

```

Ejercicio 5

Elabore un programa que solicite un número entero positivo y muestre su tabla de multiplicar del 1 al 10, además de calcular la suma de los resultados obtenidos.

```

1 //Ejercicio 5
2 //Elabore un programa que solicite un número entero positivo y
3 //muestre su tabla de multiplicar del 1 al 10,
4 //además de calcular la suma de los resultados obtenidos.
5
6 #include <iostream>
7 using namespace std;
8
9 int main(){
10     int num;
11     int suma=0;
12     cout<<"Ingrese un numero entero positivo:";
13     cin>>num;
14     if((num)>0){
15         cout<<"TABLA DE MULTIPLICAR"<<endl;
16         for(int i=1; i<=10; i++){
17             cout<<num<<" x "<<i<<" = "<<num*i<<endl;
18             suma=num*i;
19         }
20         cout<<"La suma de sus resultados son:"<<suma<<endl;
21     }
22     else{
23         cout<<"El numero ingresado no es valido.Intentelo de nuevo"<<endl;
24     }
25     return 0;
26 }
  
```

```

Ingresa un numero entero positivo:7
TABLA DE MULTIPLICAR
7x1=7
7x2=14
7x3=21
7x4=28
7x5=35
7x6=42
7x7=49
7x8=56
7x9=63
7x10=70
La suma de sus resultados son:385

-----
Process exited after 6.433 seconds with return value 0
Presione una tecla para continuar . . . |
  
```

EJERCICIOS FOR + IF

Ejercicio 6

Desarrolle un programa que muestre los números del 1 al 20 e indique para cada número si es par o impar, utilizando estructuras for e if.

```

1 //Ejercicio 6
2 //Desarrolle un programa que muestre los números del 1 al 20
3 //indique para cada número si es par o impar,
4 //utilizando estructuras for e if.
5
6 #include <iostream>
7 using namespace std;
8
9 int main(){
10     for(int i=1; i<=20; i++){
11         if(i%2==0){
12             cout<<i<<" es par"<<endl;
13         }
14         else{
15             cout<<i<<" es impar"<<endl;
16         }
17     }
18     return 0;
19 }
  
```

```

1 es impar
2 es par
3 es impar
4 es par
5 es impar
6 es par
7 es impar
8 es par
9 es impar
10 es par
11 es impar
12 es par
13 es impar
14 es par
15 es impar
16 es par
17 es impar
18 es par
19 es impar
20 es par

-----
Process exited after 0.7857 seconds with return value 0
Presione una tecla para continuar . . . |
  
```

Ejercicio 7

Elabore un programa que recorra los números del 1 al 50 y cuente cuántos son múltiplos de 3 y cuántos no lo son, mostrando ambos resultados.

The screenshot shows a C++ program in a code editor on the left and its execution output in a terminal window on the right. The program is designed to count multiples of 3 and non-multiples of 3 within the range 1 to 50.

```

1 //ejercicio 7
2 //labore un programa que recorra los números del 1 al 50 y
3 //cuente cuántos son múltiplos de 3 y cuántos no lo son,
4 //mostrando ambos resultados.
5 #include <iostream>
6 using namespace std;
7 int main()
8 {
9     int contador=0;
10    int contador1=0;
11    for(int i=1; i<=50; i++){
12        if(i%3==0){
13            cout<<i<<endl;
14            contador++;
15        }
16        else{
17            contador1++;
18        }
19        cout<<"Los numeros multiples de 3 son:"<<contador<<endl;
20        cout<<"Los numeros multiples NO de 3 son:"<<contador1<<endl;
21    }
22    return 0;
23 }
24

```

The terminal output shows the execution of the program, displaying the counts for multiples and non-multiples of 3.

```

3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
Los numeros multiples de 3 son:16
Los numeros multiples NO de 3 son:34

-----
Process exited after 0.9637 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 8

Realice un programa que permita ingresar 10 números y determine cuántos son positivos, cuántos son negativos y cuántos son iguales a cero.

The screenshot shows a C++ program in a code editor on the left and its execution output in a terminal window on the right. The program prompts the user to enter 10 numbers and then displays the count of positive, negative, and zero values.

```

4
5 #include <iostream>
6 using namespace std;
7
8 int main()
9 {
10     int numeros[10];
11     int positivo=0, negativos=0, ceros=0;
12
13     for(int i=0; i<10; i++){
14         cout<<"Ingrese el numero "<<i+1<<": ";
15         cin>>numeros[i];
16
17         if(numeros[i]>0){
18             positivo++;
19         }
20         else if(numeros[i]<0){
21             negativos++;
22         }
23         else if(numeros[i]==0){
24             ceros++;
25         }
26     }
27     cout<<"Hay "<<positivo<<" numeros positivos."<<endl;
28     cout<<"Hay "<<negativos<<" numeros negativos."<<endl;
29     cout<<"Hay "<<ceros<<" numeros iguales a cero."<<endl;
30
31     return 0;
32 }

```

The terminal output shows the user entering 10 numbers and the program's response with the counts.

```

Ingrese el numero 1:6
Ingrese el numero 2:7
Ingrese el numero 3:9
Ingrese el numero 4:0
Ingrese el numero 5:-5
Ingrese el numero 6:-3
Ingrese el numero 7:9
Ingrese el numero 8:0
Ingrese el numero 9:0
Ingrese el numero 10:-3
Hay 4 numeros positivos.
Hay 3 numeros negativos.
Hay 3 numeros iguales a cero.

-----
Process exited after 68.12 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 9

Desarrolle un programa que permita ingresar 15 números y determine cuántos valores se encuentran dentro del rango de 20 a 80 y cuántos están fuera de ese rango.

```

1 //Ejercicio 9
2 //Desarrolle un programa que permita ingresar 15 números y determine cuántos valores
3 //se encuentran dentro del rango de 20 a 80 y cuántos están fuera de ese rango.
4
5 #include <iostream>
6 using namespace std;
7
8 int main()
9 {
10     int numeros[15];
11     int rango=0, noRango=0;
12     cout<<"Ingrese 15 numeros: "<<endl;
13     for(int i=0; i<15; i++){
14         cout<<"i: "<<i<<" ";
15         cin>>numeros[i];
16     }
17     if(numeros[i]>=20 && numeros[i]<=80){
18         rango++;
19     }
20     else if(numeros[i]>20 || numeros[i]>80){
21         noRango++;
22     }
23 }
24
25 cout<<"Los numeros que pertenecen al rango son: "<<rango<<endl;
26 cout<<"Los numeros que no pertenecen al rango son: "<<noRango<<endl;
27
28 return 0;
29 }

```

Output:

```

Ingrese 15 numeros:
1:90
2:98
3:98
4:97
5:96
6:67
7:45
8:27
9:45
10:76
11:54
12:56
13:76
14:65
15:67
Los numeros que pertenecen al rango son: 10
Los numeros que no pertenecen al rango son: 5
-----
Process exited after 34.93 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 10

Elabore un programa que muestre los números del 1 al 100 y calcule la suma de los números pares y la suma de los números impares, indicando al final cuál suma es mayor.

```

1 //Ejercicio 10
2 //Elabore un programa que muestre los números del 1 al 100 y calcule la suma de los números pares
3 //y la suma de los números impares, indicando al final cuál suma es mayor.
4
5 #include <iostream>
6 using namespace std;
7
8 int main()
9 {
10     int pares=0, impares=0;
11     for(int i=1; i<=100; i++){
12         if(i%2==0){
13             pares+=i;
14         }
15         else{
16             impares+=i;
17         }
18     }
19     if(pares>impares){
20         cout<<"La suma mayor es de los numeros pares con: "<<pares<<endl;
21     }
22     else if(pares<impares){
23         cout<<"La suma mayor es de los numeros impares con: "<<impares<<endl;
24     }
25     else{
26         cout<<"Ambas sumas son iguales:"<<endl;
27     }
28
29 return 0;
30 }
31
32 }
33

```

Output:

```

La suma mayor es de los numeros pares con: 2550
-----
Process exited after 0.7593 seconds with return value 0
Presione una tecla para continuar . . .

```

ARREGLOS (ARRAYS) – CON CÁLCULOS

Ejercicio 11

Desarrolle un programa que permita ingresar **10 números en un arreglo** y calcule:

- La suma total de los valores
- El promedio
- El porcentaje de números positivos
- El porcentaje de números negativos


```

1 //Ejercicio 11
2 //Desarrolle un programa que permita ingresar 10 números en un arreglo y calcule:
3 //La suma total de los valores
4 //El promedio
5 //El porcentaje de números positivos
6 //El porcentaje de números negativos
7 #include <iostream>
8 using namespace std;
9 int main()
10 {
11     float numeros[10];
12     float suma=0;
13     float positivos=0, negativos=0;
14
15     cout<<"Ingrese 10 numeros: "<<endl;
16     for(int i=0;i<10;i++){
17         cout<<i+1<<" ";
18         cin>>numeros[i];
19         suma+=numeros[i];
20
21         if(numeros[i]>0){
22             positivos++;
23         }
24         else if(numeros[i]<0){
25             negativos++;
26         }
27     }
28     cout<<"La suma total de los valores es: "<<suma<<endl;
29     cout<<"El promedio es: "<<suma/10<<endl;
30     cout<<"Porcentaje positivos: "<<(positivos/10)*100<<endl;
31     cout<<"Porcentaje negativos: "<<(negativos/10)*100<<endl;
32
33     return 0;
34 }

```

Output:

```

Ingrese 10 numeros:
1:4
2:5
3:6
4:7
5:8
6:9
7:3
8:4
9:5
10:-8
La suma total de los valores es: 43
El promedio es: 4.3
Porcentaje positivos: 90
Porcentaje negativos: 10

-----
Process exited after 13.86 seconds with return value 0
Presione una tecla para continuar . . . |

```

Ejercicio 12

Elabore un programa que permita ingresar **8 números en un arreglo** y determine:

- El número mayor
- El número menor
- La diferencia entre el mayor y el menor

```

1 //Ejercicio 12
2 //Elabore un programa que permita ingresar 8 números en un arreglo y determine:
3 //El número mayor
4 //El número menor
5 //La diferencia entre el mayor y el menor
6
7 #include <iostream>
8 using namespace std;
9
10 int main() {
11     int numeros[8];
12     int mayor, menor;
13
14     // Ingreso de números
15     for (int i = 0; i < 8; i++) {
16         cout << "Ingrese el numero " << i + 1 << ": ";
17         cin >> numeros[i];
18     }
19
20     mayor = menor = numeros[0]; // Inicializar con el primer número
21
22     // Buscar mayor y menor
23     for (int i = 1; i < 8; i++) {
24         if (numeros[i] > mayor) mayor = numeros[i];
25         if (numeros[i] < menor) menor = numeros[i];
26     }
27
28     cout << "Numero mayor: " << mayor << endl;
29     cout << "Numero menor: " << menor << endl;
30     cout << "Diferencia entre mayor y menor: " << mayor - menor << endl;
31
32     return 0;
33 }
34

```

Output:

```

Ingrese el numero 1: 2
Ingrese el numero 2: 3
Ingrese el numero 3: 4
Ingrese el numero 4: 5
Ingrese el numero 5: 6
Ingrese el numero 6: 7
Ingrese el numero 7: 8
Ingrese el numero 8: 9
Numero mayor: 9
Numero menor: 2
Diferencia entre mayor y menor: 7

-----
Process exited after 8.802 seconds with return value 0
Presione una tecla para continuar . . . |

```

Ejercicio 13

Desarrolle un programa que permita ingresar **12 números en un arreglo** y calcule:

- La suma de los números pares
- La suma de los números impares
- Cuál suma es mayor (pares o impares)

```

1 //Ejercicio 13
2 //Desarrolle un programa que permita ingresar 12 números en un arreglo y calcule:
3 //La suma de los números pares
4 //La suma de los números impares
5 //Cuál suma es mayor (pares o impares)
6
7
8 #include <iostream>
9 using namespace std;
10
11 int main() {
12     int numeros[12];
13     int sumaPares = 0, sumaImpares = 0;
14
15     for (int i = 0; i < 12; i++) {
16         cout << "Ingrese el numero " << i + 1 << ": ";
17         cin >> numeros[i];
18         if (numeros[i] % 2 == 0)
19             sumaPares += numeros[i];
20         else
21             sumaImpares += numeros[i];
22     }
23
24     cout << "Suma de numeros pares: " << sumaPares << endl;
25     cout << "Suma de numeros impares: " << sumaImpares << endl;
26
27     if (sumaPares > sumaImpares)
28         cout << "La suma de pares es mayor." << endl;
29     else if (sumaImpares > sumaPares)
30         cout << "La suma de impares es mayor." << endl;
31     else
32         cout << "Las sumas son iguales." << endl;
33
34     return 0;
35 }

```

Ingrese el numero 1: 9
 Ingrese el numero 2: 8
 Ingrese el numero 3: 7
 Ingrese el numero 4: 6
 Ingrese el numero 5: 5
 Ingrese el numero 6: 4
 Ingrese el numero 7: 3
 Ingrese el numero 8: 2
 Ingrese el numero 9: 1
 Ingrese el numero 10: 3
 Ingrese el numero 11: 4
 Ingrese el numero 12: 5
 Suma de numeros pares: 24
 Suma de numeros impares: 33
 La suma de impares es mayor.

Process exited after 11.84 seconds with return value 0
 Presione una tecla para continuar . . .

Ejercicio 14

Realice un programa que permita ingresar **10 calificaciones (0–10)** en un arreglo y determine:

- El promedio general
- Cuántos estudiantes aprobaron (≥ 7)
- Cuántos reprobaron (< 7)

```

1 //Ejercicio 14
2 //Realice un programa que permita ingresar 10 calificaciones (0-10) en un arreglo y determine:
3 //El promedio general
4 //Cuántos estudiantes aprobaron (>= 7)
5 //Cuántos reprobaron (< 7)
6
7
8 #include <iostream>
9 using namespace std;
10
11 int main() {
12     double calificaciones[10];
13     double suma = 0;
14     int aprobados = 0, reprobados = 0;
15
16     for (int i = 0; i < 10; i++) {
17         do {
18             cout << "Ingrese la calificación " << i + 1 << " (0-10): ";
19             cin >> calificaciones[i];
20             while (calificaciones[i] < 0 || calificaciones[i] > 10);
21         } while (true);
22         suma += calificaciones[i];
23         if (calificaciones[i] >= 7) aprobados++;
24         else reprobados++;
25     }
26
27     cout << "Promedio general: " << suma / 10 << endl;
28     cout << "Cantidad de aprobados: " << aprobados << endl;
29     cout << "Cantidad de reprobados: " << reprobados << endl;
30
31     return 0;
32 }
33

```

Ingrese la calificación 1 (0-10): 5
 Ingrese la calificación 2 (0-10): 6
 Ingrese la calificación 3 (0-10): 7
 Ingrese la calificación 4 (0-10): 8
 Ingrese la calificación 5 (0-10): 5
 Ingrese la calificación 6 (0-10): 3
 Ingrese la calificación 7 (0-10): 2
 Ingrese la calificación 8 (0-10): 9
 Ingrese la calificación 9 (0-10): 10
 Ingrese la calificación 10 (0-10): 8
 Promedio general: 6.3
 Cantidad de aprobados: 5
 Cantidad de reprobados: 5

Process exited after 23.71 seconds with return value 0
 Presione una tecla para continuar . . .

Ejercicio 15

Elabore un programa que permita ingresar **15 números en un arreglo** y calcule:

- El promedio de los números positivos
- El promedio de los números negativos


```

1 //Ejercicio 15
2 //Elabore un programa que permita ingresar 15 números en un arreglo y calcule:
3 //El promedio de los números positivos
4 //El promedio de los números negativos
5 #include <iostream>
6 using namespace std;
7
8 int main() {
9     double numeros[15];
10    double sumaPos = 0, sumaNeg = 0;
11    int countPos = 0, countNeg = 0;
12
13    for (int i = 0; i < 15; i++) {
14        cout << "Ingrese el numero " << i + 1 << ": ";
15        cin >> numeros[i];
16
17        if (numeros[i] > 0) {
18            sumaPos += numeros[i];
19            countPos++;
20        } else if (numeros[i] < 0) {
21            sumaNeg += numeros[i];
22            countNeg++;
23        }
24    }
25
26    if (countPos > 0)
27        cout << "Promedio de positivos: " << sumaPos / countPos << endl;
28    else
29        cout << "No hay numeros positivos." << endl;
30
31    if (countNeg > 0)
32        cout << "Promedio de negativos: " << sumaNeg / countNeg << endl;
33    else
34        cout << "No hay numeros negativos." << endl;
35
36    return 0;
37 }

```

Ingrese el numero 1: 3
 Ingrese el numero 2: 4
 Ingrese el numero 3: 5
 Ingrese el numero 4: 6
 Ingrese el numero 5: 7
 Ingrese el numero 6: 8
 Ingrese el numero 7: 9
 Ingrese el numero 8: 0
 Ingrese el numero 9: 8
 Ingrese el numero 10: 6
 Ingrese el numero 11: 5
 Ingrese el numero 12: 4
 Ingrese el numero 13: 5
 Ingrese el numero 14: 2
 Ingrese el numero 15: 8
 Promedio de positivos: 5.71429
 No hay numeros negativos.

Process exited after 13.64 seconds with return value 0
 Presione una tecla para continuar . . . |

VECTORES – CON CÁLCULOS

Ejercicio 16

Utilizando vectores, desarrolle un programa que permita ingresar **10 números** y determine:

- La suma total
- El promedio
- Cuántos números están por encima del promedio

```

1 //Ejercicio 16
2 //Utilizando vectores, desarrolle un programa que permita ingresar 10 números y determine:
3 //La suma total
4 //El promedio
5 //Cuántos números están por encima del promedio
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10    double numeros[10];
11    double suma = 0, promedio;
12    int countArriba = 0;
13
14    for (int i = 0; i < 10; i++) {
15        cout << "Ingrese el numero " << i + 1 << ": ";
16        cin >> numeros[i];
17        suma += numeros[i];
18    }
19
20    promedio = suma / 10;
21
22    for (int i = 0; i < 10; i++) {
23        if (numeros[i] > promedio) countArriba++;
24    }
25
26    cout << "Suma total: " << suma << endl;
27    cout << "Promedio: " << promedio << endl;
28    cout << "Cantidad de numeros por encima del promedio: " << countArriba << endl;
29
30    return 0;
31 }

```

Ingrese el numero 1: 4
 Ingrese el numero 2: 5
 Ingrese el numero 3: 6
 Ingrese el numero 4: 7
 Ingrese el numero 5: 8
 Ingrese el numero 6: 9
 Ingrese el numero 7: 0
 Ingrese el numero 8: 9
 Ingrese el numero 9: 7
 Ingrese el numero 10: 4
 Suma total: 59
 Promedio: 5.9
 Cantidad de numeros por encima del promedio: 6

Process exited after 8.746 seconds with return value 0
 Presione una tecla para continuar . . . |

Ejercicio 17

Elabore un programa con vectores que permita ingresar **8 valores** y calcule:

- El valor máximo
- El valor mínimo
- El rango (máximo – mínimo)

```

1 //Ejercicio 17
2 //Elabore un programa con vectores que permita ingresar 8 valores y calcule:
3 // El valor máximo
4 // El valor mínimo
5 // El rango (máximo - mínimo)
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     int valores[8];
11     int maximo, minimo;
12     for (int i = 0; i < 8; i++) {
13         cout << "Ingrese el valor " << i + 1 << ": ";
14         cin >> valores[i];
15     }
16     maximo = minimo = valores[0];
17     for (int i = 1; i < 8; i++) {
18         if (valores[i] > maximo) maximo = valores[i];
19         if (valores[i] < minimo) minimo = valores[i];
20     }
21     cout << "Valor maximo: " << maximo << endl;
22     cout << "Valor minimo: " << minimo << endl;
23     cout << "Rango: " << maximo - minimo << endl;
24     return 0;
25 }

```

Output:

```

Ingrese el valor 1: 2
Ingrese el valor 2: 3
Ingrese el valor 3: 4
Ingrese el valor 4: 5
Ingrese el valor 5: 6
Ingrese el valor 6: 7
Ingrese el valor 7: 8
Ingrese el valor 8: 9
Valor maximo: 9
Valor minimo: 2
Rango: 7

-----
Process exited after 9.054 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 18

Desarrolle un programa que permita ingresar **12 valores en un vector** y determine:

- La suma de los valores positivos
- La suma de los valores negativos
- El valor absoluto total acumulado

```

1 //Ejercicio 18
2 //Desarrolle un programa que permita ingresar 12 valores en un vector y determine:
3 // La suma de los valores positivos
4 // La suma de los valores negativos
5 // El valor absoluto total acumulado
6 #include <iostream>
7 #include <cmath> // Para abs()
8 using namespace std;
9
10 int main() {
11     int valores[12];
12     int sumaPos = 0, sumaNeg = 0, sumaAbs = 0;
13     for (int i = 0; i < 12; i++) {
14         cout << "Ingrese el valor " << i + 1 << ": ";
15         cin >> valores[i];
16         if (valores[i] > 0) sumaPos += valores[i];
17         else if (valores[i] < 0) sumaNeg += valores[i];
18         sumaAbs += abs(valores[i]);
19     }
20     cout << "Suma de positivos: " << sumaPos << endl;
21     cout << "Suma de negativos: " << sumaNeg << endl;
22     cout << "Suma del valor absoluto total: " << sumaAbs << endl;
23     return 0;
24 }

```

Output:

```

Ingrese el valor 1: 9
Ingrese el valor 2: 8
Ingrese el valor 3: 7
Ingrese el valor 4: 6
Ingrese el valor 5: 5
Ingrese el valor 6: 4
Ingrese el valor 7: 3
Ingrese el valor 8: 2
Ingrese el valor 9: 1
Ingrese el valor 10: 0
Ingrese el valor 11: 1
Ingrese el valor 12: 2
Suma de positivos: 50
Suma de negativos: 0
Suma del valor absoluto total: 50

-----
Process exited after 19.34 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 19

Utilizando vectores, elabore un programa que permita ingresar **10 edades** y calcule:

- El promedio de edades
- Cuántas edades son mayores al promedio
- Cuántas edades son menores o iguales al promedio

```

1 //Ejercicio 19
2 //Utilizando vectores, elabore un programa que permita ingresar 10 edades y calcule:
3 //El promedio de edades
4 //Cuántas edades son mayores al promedio
5 //Cuántas edades son menores o iguales al promedio
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     int edades[10];
11     double suma = 0;
12     int mayores = 0, menoresIgual = 0;
13
14     for (int i = 0; i < 10; i++) {
15         cout << "Ingrese la edad " << i + 1 << " : ";
16         cin >> edades[i];
17         suma += edades[i];
18     }
19
20     double promedio = suma / 10;
21
22     for (int i = 0; i < 10; i++) {
23         if (edades[i] > promedio) mayores++;
24         else menoresIgual++;
25     }
26
27     cout << "Promedio de edades: " << promedio << endl;
28     cout << "Cantidad de edades mayores al promedio: " << mayores << endl;
29     cout << "Cantidad de edades menores o iguales al promedio: " << menoresIgual << endl;
30
31     return 0;
32 }

```

Output:

```

Ingrese la edad 1: 34
Ingrese la edad 2: 32
Ingrese la edad 3: 67
Ingrese la edad 4: 54
Ingrese la edad 5: 34
Ingrese la edad 6: 98
Ingrese la edad 7: 65
Ingrese la edad 8: 43
Ingrese la edad 9: 12
Ingrese la edad 10: 23
Promedio de edades: 46.2
Cantidad de edades mayores al promedio: 4
Cantidad de edades menores o iguales al promedio: 6
Process exited after 14.92 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 20

Desarrolle un programa con vectores que permita ingresar **10 números** y calcule:

- El cuadrado de cada número
- La suma de los cuadrados
- El promedio de los cuadrados

```

1 //Ejercicio 20
2 //Desarrolle un programa con vectores que permita ingresar 10 números y calcule:
3 //El cuadrado de cada número
4 //La suma de los cuadrados
5 //El promedio de los cuadrados
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     int numeros[10];
11     int sumaCuadrados = 0;
12
13     cout << "Ingrese 10 numeros:" << endl;
14     for (int i = 0; i < 10; i++) {
15         cout << "Numero " << i + 1 << " : ";
16         cin >> numeros[i];
17         sumaCuadrados += numeros[i] * numeros[i];
18     }
19
20     cout << "Cuadrados de cada numero:" << endl;
21     for (int i = 0; i < 10; i++) {
22         cout << numeros[i] << "^2 = " << numeros[i] * numeros[i] << endl;
23     }
24
25     cout << "Suma de los cuadrados: " << sumaCuadrados << endl;
26     cout << "Promedio de los cuadrados: " << sumaCuadrados / 10.0 << endl;
27
28     return 0;
29 }
30

```

Output:

```

Ingrese 10 numeros:
Numero 1: 3
Numero 2: 7
Numero 3: 8
Numero 4: 4
Numero 5: 6
Numero 6: 9
Numero 7: 2
Numero 8: 4
Numero 9: 7
Numero 10: 8
Cuadrados de cada numero:
3^2 = 9
7^2 = 49
8^2 = 64
4^2 = 16
6^2 = 36
9^2 = 81
2^2 = 4
4^2 = 16
7^2 = 49
8^2 = 64
Suma de los cuadrados: 388
Promedio de los cuadrados: 38.8
Process exited after 274.7 seconds with return value 0
Presione una tecla para continuar . . .

```

ARREGLOS / VECTORES + CONDICIONES

Ejercicio 21

Elabore un programa que permita ingresar **10 números en un arreglo** y determine:

- Cuántos están en el rango de **20 a 80**
- Cuántos están fuera de ese rango
- El porcentaje de cada grupo

```

1 //Ejercicio 21
2 //Desarrolle un programa que permita ingresar 10 números en un arreglo y determine:
3 // 1. Cuántos están en el rango de 20 a 80
4 // 2. Cuántos están fuera de ese rango
5 // 3. El porcentaje de cada grupo
6
7 #include <iostream>
8 using namespace std;
9
10 int main() {
11     int numeros[10];
12     int dentro = 0, fuera = 0;
13
14     for (int i = 0; i < 10; i++) {
15         cout << "Ingrese el numero " << i + 1 << ": ";
16         cin >> numeros[i];
17         if (numeros[i] >= 20 && numeros[i] <= 80) dentro++;
18         else fuera++;
19     }
20
21     cout << "Cantidad dentro del rango 20-80: " << dentro << endl;
22     cout << "Cantidad fuera del rango: " << fuera << endl;
23     cout << "Porcentaje dentro del rango: " << (dentro * 100.0 / 10) << "%" << endl;
24     cout << "Porcentaje fuera del rango: " << (fuera * 100.0 / 10) << "%" << endl;
25
26     return 0;
27 }
28
29

```

Output:

```

Ingrese el numero 1: 67
Ingrese el numero 2: 56
Ingrese el numero 3: 45
Ingrese el numero 4: 98
Ingrese el numero 5: 45
Ingrese el numero 6: 78
Ingrese el numero 7: 67
Ingrese el numero 8: 45
Ingrese el numero 9: 10
Ingrese el numero 10: 12
Cantidad dentro del rango 20-80: 7
Cantidad fuera del rango: 3
Porcentaje dentro del rango: 70%
Porcentaje fuera del rango: 30%

-----
Process exited after 23.51 seconds with return value
8
Presione una tecla para continuar . . .

```

Ejercicio 22

Desarrolle un programa que permita ingresar **15 valores en un vector** y calcule:

- Cuántos valores son múltiplos de 3
- Cuántos valores son múltiplos de 5
- Cuántos valores son múltiplos de ambos

```

1 //Ejercicio 22
2 //Desarrolle un programa que permita ingresar 15 valores en un vector y calcule:
3 // 1. Cuántos valores son múltiplos de 3
4 // 2. Cuántos valores son múltiplos de 5
5 // 3. Cuántos valores son múltiplos de ambos
6
7 #include <iostream>
8 using namespace std;
9
10 int main() {
11     int numeros[15];
12     int mult3 = 0, mult5 = 0, multAmbos = 0;
13
14     for (int i = 0; i < 15; i++) {
15         cout << "Ingrese el valor " << i + 1 << ": ";
16         cin >> numeros[i];
17         if (numeros[i] % 3 == 0) mult3++;
18         if (numeros[i] % 5 == 0) mult5++;
19         if (numeros[i] % 3 == 0 && numeros[i] % 5 == 0) multAmbos++;
20     }
21
22     cout << "Múltiplos de 3: " << mult3 << endl;
23     cout << "Múltiplos de 5: " << mult5 << endl;
24     cout << "Múltiplos de ambos: " << multAmbos << endl;
25
26     return 0;
27 }
28
29

```

Output:

```

Ingrese el valor 1: 2
Ingrese el valor 2: 3
Ingrese el valor 3: 4
Ingrese el valor 4: 5
Ingrese el valor 5: 6
Ingrese el valor 6: 7
Ingrese el valor 7: 8
Ingrese el valor 8: 9
Ingrese el valor 9: 0
Ingrese el valor 10: 9
Ingrese el valor 11: 8
Ingrese el valor 12: 7
Ingrese el valor 13: 6
Ingrese el valor 14: 5
Ingrese el valor 15: 3
Múltiplos de 3: 7
Múltiplos de 5: 3
Múltiplos de ambos: 1

-----
Process exited after 24.03 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 23

Elabore un programa que permita ingresar **10 precios** y calcule:

- El subtotal
- El IVA (12 %)
- El total a pagar

```

1 //Ejercicio 23
2 //Elabore un programa que permita ingresar 10 precios y calcule:
3 // El subtotal
4 // El IVA (12 %)
5 // El total a pagar
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     double precios[10];
11     double subtotal = 0, iva, total;
12
13     for (int i = 0; i < 10; i++) {
14         cout << "Ingrese el precio " << i + 1 << ": ";
15         cin >> precios[i];
16         subtotal += precios[i];
17     }
18
19     iva = subtotal * 0.12;
20     total = subtotal + iva;
21
22     cout << "Subtotal: " << subtotal << endl;
23     cout << "IVA (12%): " << iva << endl;
24     cout << "Total a pagar: " << total << endl;
25
26     return 0;
27 }

```

```

Ingrese el precio 1: 23
Ingrese el precio 2: 45
Ingrese el precio 3: 67
Ingrese el precio 4: 87
Ingrese el precio 5: 34
Ingrese el precio 6: 45
Ingrese el precio 7: 23
Ingrese el precio 8: 21
Ingrese el precio 9: 12
Ingrese el precio 10: 15
Subtotal: 372
IVA (12%): 44.64
Total a pagar: 416.64

-----
Process exited after 14.93 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 24

Desarrolle un programa que permita ingresar **10 sueldos** y calcule:

- El sueldo promedio
- Cuántos sueldos están por encima del promedio
- Cuántos están por debajo del promedio

```

1 //Ejercicio 24
2 //Desarrolle un programa que permita ingresar 10 sueldos y calcule:
3 // El sueldo promedio
4 // Cuántos sueldos están por encima del promedio
5 // Cuántos están por debajo del promedio
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     double sueldos[10];
11     double suma = 0;
12
13     for (int i = 0; i < 10; i++) {
14         cout << "Ingrese el sueldo " << i + 1 << ": ";
15         cin >> sueldos[i];
16         suma += sueldos[i];
17     }
18
19     double promedio = suma / 10;
20     int arriba = 0, debajo = 0;
21
22     for (int i = 0; i < 10; i++) {
23         if (sueldos[i] > promedio) arriba++;
24         else if (sueldos[i] < promedio) debajo++;
25     }
26
27     cout << "Sueldo promedio: " << promedio << endl;
28     cout << "Cantidad de sueldos arriba del promedio: " << arriba << endl;
29     cout << "Cantidad de sueldos debajo del promedio: " << debajo << endl;
30
31     return 0;
32 }

```

```

Ingrese el sueldo 1: 200
Ingrese el sueldo 2: 300
Ingrese el sueldo 3: 546
Ingrese el sueldo 4: 434
Ingrese el sueldo 5: 567
Ingrese el sueldo 6: 432
Ingrese el sueldo 7: 234
Ingrese el sueldo 8: 567
Ingrese el sueldo 9: 543
Ingrese el sueldo 10: 543
Sueldo promedio: 436.6
Cantidad de sueldos arriba del promedio: 5
Cantidad de sueldos debajo del promedio: 5

-----
Process exited after 51.19 seconds with return value 0
Presione una tecla para continuar . . .

```

RETO (INTEGRADOR)

Ejercicio 25

Desarrolle un programa que permita ingresar **10 notas finales** y calcule:

- El promedio general
- La nota más alta y la más baja
- El porcentaje de aprobados y reprobados
- Muestre un resumen final de resultados


```

10 int main() {
11     double notas[10];
12     double suma = 0;
13     double mayor, menor;
14     int aprobados = 0, reprobados = 0;
15
16     for (int i = 0; i < 10; i++) {
17         do {
18             cout << "Ingrese la nota " << i + 1 << " (0-10): ";
19             cin >> notas[i];
20             while (notas[i] < 0 || notas[i] > 10);
21             suma += notas[i];
22         }
23
24         mayor = menor = notas[0];
25
26         for (int i = 0; i < 10; i++) {
27             if (notas[i] >= 7) aprobados++;
28             else reprobados++;
29
30             if (notas[i] > mayor) mayor = notas[i];
31             if (notas[i] < menor) menor = notas[i];
32         }
33
34         double promedio = suma / 10;
35         double porcAprob = aprobados * 10.0; // 10% por cada estudiante
36         double porcReprob = reprobados * 10.0;
37
38         cout << "\n--- RESUMEN FINAL ---" << endl;
39         cout << "Promedio general: " << promedio << endl;
40         cout << "Nota mayor: " << mayor << endl;
41         cout << "Nota menor: " << menor << endl;
42         cout << "Porcentaje de aprobados: " << porcAprob << "%" << endl;
43         cout << "Porcentaje de reprobados: " << porcReprob << "%" << endl;
44
45         return 0;
46     }
47 }

```

```

C:\Users\USUARIO\OneDrive\
Ingrese la nota 1 (0-10): 9
Ingrese la nota 2 (0-10): 7
Ingrese la nota 3 (0-10): 8
Ingrese la nota 4 (0-10): 9
Ingrese la nota 5 (0-10): 8
Ingrese la nota 6 (0-10): 7
Ingrese la nota 7 (0-10): 6
Ingrese la nota 8 (0-10): 5
Ingrese la nota 9 (0-10): 7
Ingrese la nota 10 (0-10): 8

--- RESUMEN FINAL ---
Promedio general: 7.4
Nota mayor: 9
Nota menor: 5
Porcentaje de aprobados: 80%
Porcentaje de reprobados: 20%

Process exited after 10.54 seconds with return value 0
Presione una tecla para continuar . . .

```

MENÚ

Ejercicio 26

Desarrolle un programa que muestre un menú con las siguientes opciones:

1. Ingresar dos números
2. Mostrar la suma y la resta
3. Mostrar el mayor de los dos números
4. Salir

```

1 //2. Mostrar la suma y la resta
2 //3. Mostrar el mayor de los dos números
3 //4. Salir
4
5 #include <iostream>
6 using namespace std;
7
8 int main() {
9     int opcion;
10    double num1 = 0, num2 = 0;
11
12    do {
13        cout << "\n--- MENU ---\n";
14        cout << "1. Ingresar dos numeros\n";
15        cout << "2. Mostrar suma y resta\n";
16        cout << "3. Mostrar el mayor de los dos numeros\n";
17        cout << "4. Salir\n";
18        cout << "Ingrese una opcion: ";
19        cin >> opcion;
20
21        switch(opcion) {
22            case 1:
23                cout << "Ingrese el primer numero: ";
24                cin >> num1;
25                cout << "Ingrese el segundo numero: ";
26                cin >> num2;
27                break;
28            case 2:
29                cout << "Suma: " << num1 + num2 << endl;
30                cout << "Resta: " << num1 - num2 << endl;
31                break;
32            case 3:
33                if (num1 > num2) cout << "El mayor es: " << num1 << endl;
34                else if (num2 > num1) cout << "El mayor es: " << num2 << endl;
35                else cout << "Ambos numeros son iguales." << endl;
36                break;
37            case 4:
38                cout << "Saliendo del programa..." << endl;
39                break;
40            default:
41                cout << "opcion no valida." << endl;
42        }
43    } while(opcion != 4);
44
45    return 0;
46 }

```

```

C:\Users\USUARIO\OneD
--- MENU ---
1. Ingresar dos numeros
2. Mostrar suma y resta
3. Mostrar el mayor de los dos numeros
4. Salir
Ingrese una opcion: 1
Ingrese el primer numero: 2
Ingrese el segundo numero: 3

--- MENU ---
1. Ingresar dos numeros
2. Mostrar suma y resta
3. Mostrar el mayor de los dos numeros
4. Salir
Ingrese una opcion: 2
Suma: 5
Resta: -1

--- MENU ---
1. Ingresar dos numeros
2. Mostrar suma y resta
3. Mostrar el mayor de los dos numeros
4. Salir
Ingrese una opcion: 3
El mayor es: 3

--- MENU ---
1. Ingresar dos numeros
2. Mostrar suma y resta

```

Ejercicio 27

Elabore un programa que muestre un menú con las siguientes opciones:

1. Ingresar un número
2. Verificar si el número está en el rango de 1 a 100
3. Verificar si el número es par o impar
4. Salir


```

1 //1. Verificar si el número es par o impar
2 //2. Salir
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     int opcion;
8     int numero = 0;
9
10    do {
11        cout << "n--- MENU ---\n";
12        cout << "1. Ingresar un número\n";
13        cout << "2. Verificar si el número está en el rango 1-100\n";
14        cout << "3. Verificar si el número es par o impar\n";
15        cout << "4. Salir\n";
16        cout << "Ingrese una opción: ";
17        cin >> opcion;
18
19        switch(opcion) {
20            case 1:
21                cout << "Ingrese el número: ";
22                cin >> numero;
23                break;
24            case 2:
25                if (numero >= 1 && numero <= 100)
26                    cout << "El número está en el rango 1-100." << endl;
27                else
28                    cout << "El número está fuera del rango." << endl;
29                break;
30            case 3:
31                if (numero % 2 == 0) cout << "El número es par." << endl;
32                else cout << "El número es impar." << endl;
33                break;
34            case 4:
35                cout << "Saliendo del programa..." << endl;
36                break;
37            default:
38                cout << "Opción no válida." << endl;
39        }
40    } while(opcion != 4);
41
42    return 0;
43 }

```

Errors: 0
Warnings: 0
Output Filename: C:\Users\USUARIO\OneDrive\Documents\E27.exe
Output Size: 13.799699223022 MiB
Compilation Time: 2.88s

Ejercicio 28

Desarrolle un programa que muestre un menú con las siguientes opciones:

1. Ingresar 10 números en un arreglo
2. Mostrar el promedio
3. Mostrar el mayor y el menor
4. Contar números positivos y negativos
5. Salir

```

1 //1. Ingresar 10 números en un arreglo
2 //2. Mostrar el promedio
3 //3. Mostrar el mayor y el menor
4 //4. Contar números positivos y negativos
5 //5. Salir
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     int opcion;
11     int numeros[10];
12     bool ingresado = false;
13
14     do {
15         cout << "n--- MENU ---\n";
16         cout << "1. Ingresar 10 números en un arreglo\n";
17         cout << "2. Mostrar el promedio\n";
18         cout << "3. Mostrar el mayor y el menor\n";
19         cout << "4. Contar números positivos y negativos\n";
20         cout << "5. Salir\n";
21         cout << "Ingrese una opción: ";
22         cin >> opcion;
23
24         switch(opcion) {
25             case 1:
26                 for (int i = 0; i < 10; i++) {
27                     cout << "Ingrese el número " << i + 1 << ": ";
28                     cin >> numeros[i];
29                     ingresado = true;
30                 }
31                 break;
32             case 2:
33                 if (!ingresado) cout << "Primero ingrese los números." << endl;
34                 else {
35                     double suma = 0;
36                     for (int i = 0; i < 10; i++) suma += numeros[i];
37                     cout << "Promedio: " << suma / 10 << endl;
38                 }
39                 break;
40             case 3:
41                 if (!ingresado) cout << "Primero ingrese los números." << endl;
42                 else {
43                     int maximo = numeros[0], minimo = numeros[0];
44                     for (int i = 1; i < 10; i++) {
45                         if (numeros[i] > maximo) maximo = numeros[i];
46                         if (numeros[i] < minimo) minimo = numeros[i];
47                     }
48                     cout << "Mayor: " << maximo << ", Menor: " << minimo << endl;
49                 }
50                 break;
51             case 4:
52                 if (!ingresado) cout << "Primero ingrese los números." << endl;
53                 else {
54                     int positivos = 0, negativos = 0;
55                     for (int i = 0; i < 10; i++) {
56

```

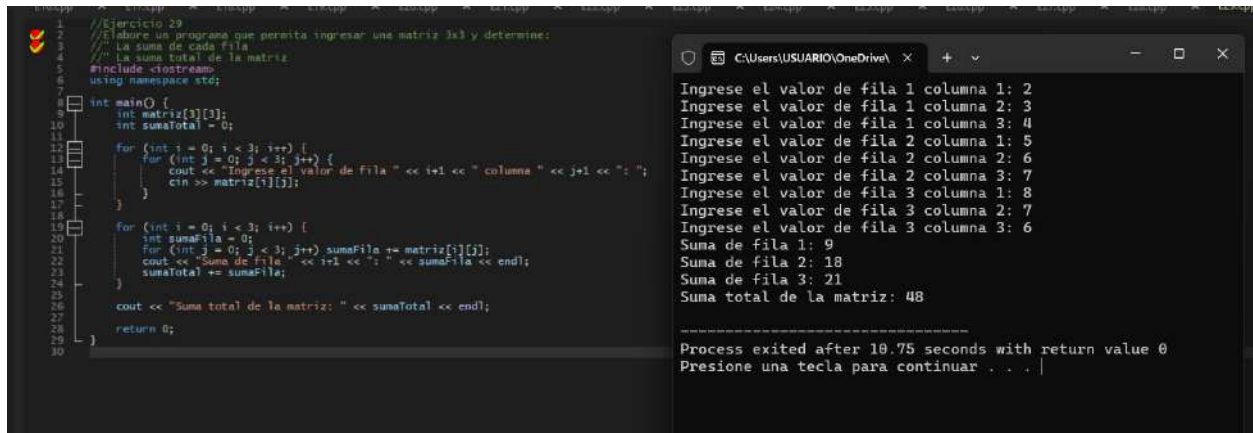
Errors: 0
Warnings: 0
Output Filename: C:\Users\USUARIO\OneDrive\Documents\E28.exe
Output Size: 13.8011617660522 MiB
Compilation Time: 2.27s

F. MATRICES

Ejercicio 29

Elabore un programa que permita ingresar una **matriz 3x3** y determine:

- La suma de cada fila
- La suma total de la matriz



```

1 //Ejercicio 29
2 //Elabore un programa que permita ingresar una matriz 3x3 y determine:
3 // La suma de cada fila
4 // La suma total de la matriz
5 #include <iostream>
6 using namespace std;
7
8 int main() {
9     int matriz[3][3];
10    int sumaTotal = 0;
11
12    for (int i = 0; i < 3; i++) {
13        for (int j = 0; j < 3; j++) {
14            cout << "Ingrese el valor de fila " << i+1 << " columna " << j+1 << ": ";
15            cin >> matriz[i][j];
16        }
17    }
18
19    for (int i = 0; i < 3; i++) {
20        int sumaFila = 0;
21        for (int j = 0; j < 3; j++) sumaFila += matriz[i][j];
22        cout << "Suma de fila " << i+1 << ": " << sumaFila << endl;
23        sumaTotal += sumaFila;
24    }
25
26    cout << "Suma total de la matriz: " << sumaTotal << endl;
27    return 0;
28 }

```

Output:

```

Ingrese el valor de fila 1 columna 1: 2
Ingrese el valor de fila 1 columna 2: 3
Ingrese el valor de fila 1 columna 3: 4
Ingrese el valor de fila 2 columna 1: 5
Ingrese el valor de fila 2 columna 2: 6
Ingrese el valor de fila 2 columna 3: 7
Ingrese el valor de fila 3 columna 1: 8
Ingrese el valor de fila 3 columna 2: 7
Ingrese el valor de fila 3 columna 3: 6
Suma de fila 1: 9
Suma de fila 2: 18
Suma de fila 3: 21
Suma total de la matriz: 48

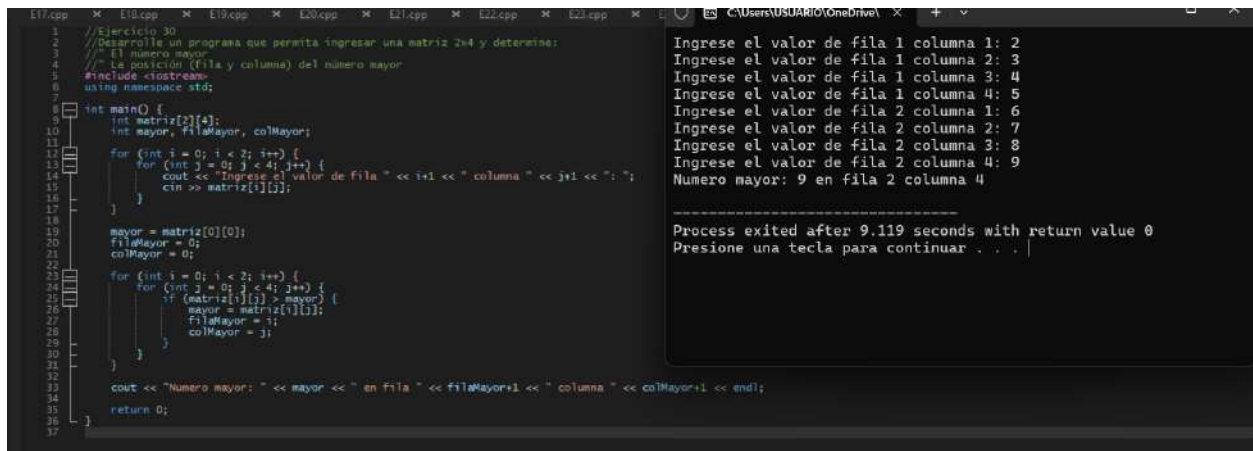
```

Process exited after 10.75 seconds with return value 0
Presione una tecla para continuar . . .

Ejercicio 30

Desarrolle un programa que permita ingresar una **matriz 2x4** y determine:

- El número mayor
- La posición (fila y columna) del número mayor



```

1 //Ejercicio 30
2 //Desarrolle un programa que permita ingresar una matriz 2x4 y determine:
3 // El número mayor
4 // La posición (fila y columna) del número mayor
5 #include <iostream>
6 using namespace std;
7
8 int main() {
9     int matriz[2][4];
10    int mayor, filaMayor, colMayor;
11
12    for (int i = 0; i < 2; i++) {
13        for (int j = 0; j < 4; j++) {
14            cout << "Ingrese el valor de fila " << i+1 << " columna " << j+1 << ": ";
15            cin >> matriz[i][j];
16        }
17    }
18
19    mayor = matriz[0][0];
20    filaMayor = 0;
21    colMayor = 0;
22
23    for (int i = 0; i < 2; i++) {
24        for (int j = 0; j < 4; j++) {
25            if (matriz[i][j] > mayor) {
26                mayor = matriz[i][j];
27                filaMayor = i;
28                colMayor = j;
29            }
30        }
31    }
32
33    cout << "Numero mayor: " << mayor << " en fila " << filaMayor+1 << " columna " << colMayor+1 << endl;
34    return 0;
35 }

```

Output:

```

Ingrese el valor de fila 1 columna 1: 2
Ingrese el valor de fila 1 columna 2: 3
Ingrese el valor de fila 1 columna 3: 4
Ingrese el valor de fila 1 columna 4: 5
Ingrese el valor de fila 2 columna 1: 6
Ingrese el valor de fila 2 columna 2: 7
Ingrese el valor de fila 2 columna 3: 8
Ingrese el valor de fila 2 columna 4: 9
Numero mayor: 9 en fila 2 columna 4

```

Process exited after 9.119 seconds with return value 0
Presione una tecla para continuar . . .

Ejercicio 31

Realice un programa que permita ingresar una **matriz 4x4** y cuente cuántos valores son:

- Positivos
- Negativos
- Iguales a cero

```

1 //Ejercicio 31
2 //Realice un programa que permita ingresar una matriz 4x4 y cuente cuántos valores son:
3 // Positivos
4 // Negativos
5 // Iguales a cero
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     int matriz[4][4];
11     int positivos = 0, negativos = 0, ceros = 0;
12
13     for (int i = 0; i < 4; i++) {
14         for (int j = 0; j < 4; j++) {
15             cout << "Ingrese el valor de fila " << i+1 << " columna " << j+1 << ": ";
16             cin >> matriz[i][j];
17
18             if (matriz[i][j] > 0) positivos++;
19             else if (matriz[i][j] < 0) negativos++;
20             else ceros++;
21         }
22     }
23
24     cout << "Cantidad de positivos: " << positivos << endl;
25     cout << "Cantidad de negativos: " << negativos << endl;
26     cout << "Cantidad de ceros: " << ceros << endl;
27
28     return 0;
29 }
30

```

Ingrese el valor de fila 1 columna 1: 2
 Ingrese el valor de fila 1 columna 2: 7
 Ingrese el valor de fila 1 columna 3: 6
 Ingrese el valor de fila 1 columna 4: 5
 Ingrese el valor de fila 2 columna 1: 9
 Ingrese el valor de fila 2 columna 2: 3
 Ingrese el valor de fila 2 columna 3: 6
 Ingrese el valor de fila 2 columna 4: 5
 Ingrese el valor de fila 3 columna 1: 3
 Ingrese el valor de fila 3 columna 2: 8
 Ingrese el valor de fila 3 columna 3: 6
 Ingrese el valor de fila 3 columna 4: 5
 Ingrese el valor de fila 4 columna 1: 4
 Ingrese el valor de fila 4 columna 2: 5
 Ingrese el valor de fila 4 columna 3: 4
 Ingrese el valor de fila 4 columna 4: 3
 Cantidad de positivos: 16
 Cantidad de negativos: 0
 Cantidad de ceros: 0

Process exited after 19.95 seconds with return value 0
 Presione una tecla para continuar . . .

EJERCICIO 32: struct + vector

Enunciado:

Desarrolle un programa en C++ que defina una estructura Cliente con los campos nombre y edad. El programa debe permitir ingresar varios clientes usando un vector y luego mostrar en pantalla la información de todos los clientes registrados.

```

1 //EJERCICIO 32: struct + vector
2 //Enunciado:
3 //Desarrolle un programa en C++ que defina una estructura Cliente con los campos nombre y edad.
4 //El programa debe permitir ingresar varios clientes usando un vector y luego mostrar en pantalla la información de todos los clientes registrados.
5 #include <iostream>
6 #include <vector>
7 using namespace std;
8
9 struct Cliente {
10     string nombre;
11     int edad;
12 };
13
14 int main() {
15     vector<Cliente> clientes;
16     int n;
17     cout << "Cuántos clientes desea ingresar? ";
18     cin >> n;
19
20     for (int i = 0; i < n; i++) {
21         Cliente c;
22         cout << "Nombre del cliente " << i+1 << ": ";
23         cin >> c.nombre;
24         cout << "Edad del cliente " << i+1 << ": ";
25         cin >> c.edad;
26         clientes.push_back(c);
27     }
28
29     cout << "\n--- Clientes Registrados ---\n";
30     for (int i = 0; i < clientes.size(); i++) {
31         cout << "Nombre: " << clientes[i].nombre << ", Edad: " << clientes[i].edad << endl;
32     }
33
34     return 0;
35 }
36

```

Cuántos clientes desea ingresar? 2
 Nombre del cliente 1: KAREN
 Edad del cliente 1: 21
 Nombre del cliente 2: LUIS
 Edad del cliente 2: 20

--- Clientes Registrados ---
 Nombre: KAREN, Edad: 21
 Nombre: LUIS, Edad: 20

Process exited after 21.6 seconds with return value 0
 Presione una tecla para continuar . . .

- EJERCICIO 33: struct + array

Enunciado:

Cree un programa en C++ que utilice una estructura Cliente con los campos nombre y edad. Almacene los datos de tres clientes en un arreglo y muestre únicamente los clientes que sean mayores de edad (edad mayor o igual a 18).

```

1 // EJERCICIO 33: struct + array
2 //Enunciado:
3 //Cree un programa en C++ que utilice una estructura Cliente con los campos nombre y edad.
4 //Almacene los datos de tres clientes en un arreglo y muestre únicamente los clientes que sean mayores de 20 años.
5 #include <iostream>
6 using namespace std;
7
8 struct Cliente {
9     string nombre;
10    int edad;
11};
12
13 int main() {
14     Cliente clientes[3];
15
16     for (int i = 0; i < 3; i++) {
17         cout << "Nombre del cliente " << i+1 << ": ";
18         cin >> clientes[i].nombre;
19         cout << "Edad del cliente " << i+1 << ": ";
20         cin >> clientes[i].edad;
21     }
22
23     cout << "\n--- Clientes Mayores de Edad ---\n";
24     for (int i = 0; i < 3; i++) {
25         if (clientes[i].edad >= 20) {
26             cout << "Nombre: " << clientes[i].nombre << ", Edad: " << clientes[i].edad << endl;
27         }
28     }
29     return 0;
30 }

```

Nombre del cliente 1: KAREN
Edad del cliente 1: 21
Nombre del cliente 2: LUIS
Edad del cliente 2: 20
Nombre del cliente 3: CHRISTIAN
Edad del cliente 3: 17

--- Clientes Mayores de Edad ---
Nombre: KAREN, Edad: 21
Nombre: LUIS, Edad: 20

Process exited after 91.69 seconds with return value 0
Presione una tecla para continuar . . . |

• EJERCICIO 34: struct + vector + archivo

Enunciado:

Realice un programa en C++ que use una estructura Cliente con los campos nombre y edad. El programa debe almacenar los datos en un vector, guardarlos en un archivo de texto y permitir buscar un cliente por su nombre, mostrando su edad si se encuentra registrado.

```

1 // EJERCICIO 34: struct + vector + archivo
2 //Enunciado:
3 //Realice un programa en C++ que use una estructura Cliente con los campos nombre y edad.
4 //El programa debe almacenar los datos en un vector, guardarlos en un archivo de texto y permitir buscar un cliente por su nombre,
5 //mostrando su edad si se encuentra registrado.
6 #include <iostream>
7 #include <vector>
8 #include <fstream>
9 using namespace std;
10
11 struct Cliente {
12     string nombre;
13     int edad;
14 };
15
16 int main() {
17     vector<Cliente> clientes;
18     int n;
19     cout << "Cuantos clientes desea ingresar? ";
20     cin >> n;
21
22     for (int i = 0; i < n; i++) {
23         Cliente c;
24         cout << "Nombre del cliente " << i+1 << ": ";
25         cin >> c.nombre;
26         cout << "Edad del cliente " << i+1 << ": ";
27         cin >> c.edad;
28         clientes.push_back(c);
29     }
30
31     // Guardar en archivo
32     ofstream archivo("clientes.txt");
33     for (int i = 0; i < clientes.size(); i++) {
34         archivo << clientes[i].nombre << " " << clientes[i].edad << endl;
35     }
36     archivo.close();
37
38     // Buscar cliente
39     string buscar;
40     cout << "Ingrese el nombre del cliente a buscar: ";

```

Cuantos clientes desea ingresar? 3
Nombre del cliente 1: KAREN
Edad del cliente 1: 21
Nombre del cliente 2: LUIS
Edad del cliente 2: 20
Nombre del cliente 3: CHRISTIAN
Edad del cliente 3: 19
Ingrese el nombre del cliente a buscar: LUIS
Edad del cliente: 20

Process exited after 43.43 seconds with return value 0
Presione una tecla para continuar . . . |

• EJERCICIO 35: struct + vector + archivo + cálculos

Enunciado:

Desarrolle un programa en C++ que defina una estructura Cliente con los campos nombre, edad y consumoMensual. El programa debe almacenar los datos en un vector, guardarlos en un archivo, leerlos posteriormente y calcular el consumo total, el consumo promedio y el cliente con mayor consumo.


```

7 #include <iostream>
8 #include <vector>
9 #include <fstream>
10 using namespace std;
11
12 struct Cliente {
13     string nombre;
14     int edad;
15     double consumoMensual;
16 };
17
18 int main() {
19     vector<Cliente> clientes;
20     int n;
21     cout << "Cuantos clientes desea ingresar? ";
22     cin >> n;
23
24     for (int i = 0; i < n; i++) {
25         Cliente c;
26         cout << "Nombre del cliente " << i+1 << ": ";
27         cin >> c.nombre;
28         cout << "Edad del cliente " << i+1 << ": ";
29         cin >> c.edad;
30         cout << "Consumo mensual: ";
31         cin >> c.consumoMensual;
32         clientes.push_back(c);
33     }
34
35     // Guardar en archivo
36     ofstream archivo("clientes.txt");
37     for (int i = 0; i < clientes.size(); i++) {
38         archivo << clientes[i].nombre << " " << clientes[i].edad << " " << clientes[i].consumoMensual << endl;
39     }
40     archivo.close();
41
42     // Leer desde archivo
43     vector<Cliente> clientesArchivo;
44     ifstream archivoLeer("clientes.txt");
45     string nombre;
46     int edad;
47     double consumoMensual;
48
49     while (archivoLeer >> nombre && archivoLeer >> edad && archivoLeer >> consumoMensual) {
50         Cliente c;
51         c.nombre = nombre;
52         c.edad = edad;
53         c.consumoMensual = consumoMensual;
54         clientesArchivo.push_back(c);
55     }
56     archivoLeer.close();
57
58     // Calcular totales y promedios
59     double totalConsumo = 0;
60     double promedioConsumo = 0;
61     string mayorConsumoNombre;
62     double mayorConsumo = 0;
63
64     for (int i = 0; i < clientesArchivo.size(); i++) {
65         totalConsumo += clientesArchivo[i].consumoMensual;
66         promedioConsumo += clientesArchivo[i].consumoMensual;
67         if (clientesArchivo[i].consumoMensual > mayorConsumo) {
68             mayorConsumo = clientesArchivo[i].consumoMensual;
69             mayorConsumoNombre = clientesArchivo[i].nombre;
70         }
71     }
72     promedioConsumo /= clientesArchivo.size();
73
74     cout << "Total de consumo: " << totalConsumo << endl;
75     cout << "Promedio de consumo: " << promedioConsumo << endl;
76     cout << "Cliente con mayor consumo: " << mayorConsumoNombre << " (" << mayorConsumo << ")" << endl;
77
78     cout << "-----" << endl;
79     cout << "Process exited after 21.79 seconds with return value 0" << endl;
80     cout << "Presione una tecla para continuar . . . |" << endl;
81     cin.get();
82 }

```

- EJERCICIO 36: struct + archivo + búsqueda y cálculos

Enunciado:

Cree un programa en C++ que utilice una estructura Estudiante con los campos nombre, nota1, nota2 y nota3. El programa debe guardar los datos en un archivo, leerlos, calcular el promedio de cada estudiante y permitir buscar un estudiante por nombre para mostrar sus notas, su promedio y si aprueba o reprueba (promedio mayor o igual a 7).

```

4 //El programa debe guardar los datos en un archivo, leerlos, calcular el promedio de cada estudiante y
5 //permitir buscar un estudiante por nombre para mostrar sus notas, su promedio y si aprueba o reprueba (promedio mayor o igual a 7).
6 #include <iostream>
7 #include <fstream>
8 #include <vector>
9 using namespace std;
10
11 struct Estudiante {
12     string nombre;
13     double nota1, nota2, nota3;
14 };
15
16 int main() {
17     vector<Estudiante> estudiantes;
18     int n;
19     cout << "Cuantos estudiantes desea ingresar? ";
20     cin >> n;
21
22     for (int i = 0; i < n; i++) {
23         Estudiante e;
24         cout << "Nombre del estudiante " << i+1 << ": ";
25         cin >> e.nombre;
26         cout << "Nota 1: "; cin >> e.nota1;
27         cout << "Nota 2: "; cin >> e.nota2;
28         cout << "Nota 3: "; cin >> e.nota3;
29         estudiantes.push_back(e);
30     }
31
32     // Guardar en archivo
33     ofstream archivo("estudiantes.txt");
34     for (int i = 0; i < estudiantes.size(); i++) {
35         archivo << estudiantes[i].nombre << " " << estudiantes[i].nota1 << " " << estudiantes[i].nota2 << " " << estudiantes[i].nota3 << endl;
36     }
37     archivo.close();
38
39     // Leer desde archivo
40     vector<Estudiante> lista;
41     ifstream archivoLeer("estudiantes.txt");
42     string nombre;
43     double nota1, nota2, nota3;
44
45     while (archivoLeer >> nombre && archivoLeer >> nota1 && archivoLeer >> nota2 && archivoLeer >> nota3) {
46         Estudiante e;
47         e.nombre = nombre;
48         e.nota1 = nota1;
49         e.nota2 = nota2;
50         e.nota3 = nota3;
51         lista.push_back(e);
52     }
53     archivoLeer.close();
54
55     // Calcular promedios y estado
56     double promedio;
57     string estado;
58
59     for (int i = 0; i < lista.size(); i++) {
60         promedio = (lista[i].nota1 + lista[i].nota2 + lista[i].nota3) / 3;
61         estado = (promedio >= 7) ? "Aprobado" : "Reprobado";
62         cout << "Nombre del estudiante " << i+1 << ": " << lista[i].nombre << endl;
63         cout << "Nota 1: " << lista[i].nota1 << " Nota 2: " << lista[i].nota2 << " Nota 3: " << lista[i].nota3 << endl;
64         cout << "Promedio: " << promedio << " Estado: " << estado << endl;
65     }
66
67     cout << "-----" << endl;
68     cout << "Process exited after 24.71 seconds with return value 0" << endl;
69     cout << "Presione una tecla para continuar . . . |" << endl;
70     cin.get();
71 }

```

- EJERCICIO 37: funciones + paso por referencia + vector

Enunciado:

Desarrolle un programa en C++ que utilice funciones con paso de parámetros por referencia y un vector de números enteros. El programa debe incluir funciones para llenar el vector y para calcular la suma, el promedio, el valor mayor y el valor menor de los elementos almacenados.

```

1 //El programa debe incluir funciones para llenar el vector y para calcular la suma, el promedio, el valor mayor y el valor menor de los elementos almacenados.
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5
6 // Funcion para llenar el vector
7 void llenarVector(vector<int> &v) {
8     for (int i = 0; i < v.size(); i++) {
9         cout << "Ingrese el numero " << i+1 << " : ";
10        cin >> v[i];
11    }
12 }
13
14 // Funcion para calcular suma y promedio
15 void calcularSumaPromedio(const vector<int> &v, int &suma, double &promedio) {
16     suma = 0;
17     for (int i = 0; i < v.size(); i++) suma += v[i];
18     promedio = suma / double(v.size());
19 }
20
21 // Funcion para calcular mayor y menor
22 void calcularMayorMenor(const vector<int> &v, int &mayor, int &menor) {
23     mayor = menor = v[0];
24     for (int i = 1; i < v.size(); i++) {
25         if (v[i] > mayor) mayor = v[i];
26         if (v[i] < menor) menor = v[i];
27     }
28 }
29
30 int main() {
31     vector<int> numeros(5); // Cambiar tamaño si se desea
32     int suma, mayor, menor;
33     double promedio;
34
35     llenarVector(numeros);
36     calcularSumaPromedio(numeros, suma, promedio);
37     calcularMayorMenor(numeros, mayor, menor);
38
39     cout << "Suma: " << suma << endl;
40     cout << "Promedio: " << promedio << endl;
41 }

```

```

Ingrese el numero 1: 4
Ingrese el numero 2: 5
Ingrese el numero 3: 6
Ingrese el numero 4: 7
Ingrese el numero 5: 8
Suma: 30
Promedio: 6
Mayor: 8, Menor: 4

-----
Process exited after 8.653 seconds with return value 0
Presione una tecla para continuar . . .

```

• EJERCICIO 38: funciones + paso por referencia + array

Enunciado:

Cree un programa en C++ que utilice un arreglo de números reales y funciones con paso por referencia. El programa debe llenar el arreglo mediante una función y calcular el total de ventas, el promedio de ventas y cuántas ventas son mayores al promedio.

```

1 // EJERCICIO 38: Funciones + paso por referencia + array
2 //Enunciado:
3 //Cree un programa en C++ que utilice un arreglo de números reales y funciones con paso por referencia.
4 //El programa debe llenar el arreglo mediante una función y calcular el total de ventas, el promedio de ventas y cuántas ventas son mayores al promedio.
5 #include <iostream>
6 using namespace std;
7
8 void llenarArreglo(double arr[], int n) {
9     for (int i = 0; i < n; i++) {
10        cout << "Ingrese la venta " << i+1 << " : ";
11        cin >> arr[i];
12    }
13 }
14
15 void calcularTotalPromedio(const double arr[], int n, double &total, double &promedio) {
16     total = 0;
17     for (int i = 0; i < n; i++) total += arr[i];
18     promedio = total / n;
19 }
20
21 int contarMayorPromedio(const double arr[], int n, double promedio) {
22     int count = 0;
23     for (int i = 0; i < n; i++) {
24         if (arr[i] > promedio) count++;
25     }
26     return count;
27 }
28
29 int main() {
30     const int N = 5; // Cambiar tamaño si se desea
31     double ventas[N];
32     double total, promedio;
33
34     llenarArreglo(ventas, N);
35     calcularTotalPromedio(ventas, N, total, promedio);
36     int mayores = contarMayorPromedio(ventas, N, promedio);
37
38     cout << "Total de ventas: " << total << endl;
39     cout << "Promedio de ventas: " << promedio << endl;
40     cout << "Ventas mayores al promedio: " << mayores << endl;
41 }

```

```

Ingrese la venta 1: 23
Ingrese la venta 2: 34
Ingrese la venta 3: 32
Ingrese la venta 4: 54
Ingrese la venta 5: 67
Total de ventas: 210
Promedio de ventas: 42
Ventas mayores al promedio: 2

-----
Process exited after 9.256 seconds with return value 0
Presione una tecla para continuar . . .

```

• EJERCICIO 39: funciones + paso por valor + vector

Enunciado:

Desarrolle un programa en C++ que utilice funciones con paso de parámetros por valor y un vector de números enteros. El programa debe pasar el vector a las funciones para calcular la suma de los elementos, el promedio y determinar cuántos valores son pares e impares, sin modificar el contenido original del vector.


```

4 //El programa debe pasar el vector a las funciones para calcular la suma de los elementos, el promedio y determinar cuántos valores son pares e impares.
5 //sin modificar el contenido original del vector.
6 #include <iostream>
7 #include <vector>
8 using namespace std;
9
10 int calcularSuma(vector<int> v) {
11     int suma = 0;
12     for (int i = 0; i < v.size(); i++) suma += v[i];
13     return suma;
14 }
15
16 double calcularPromedio(vector<int> v) {
17     return calcularSuma(v) / double(v.size());
18 }
19
20 void contarParesImpares(vector<int> v, int &pares, int &impares) {
21     pares = impares = 0;
22     for (int i = 0; i < v.size(); i++) {
23         if (v[i] % 2 == 0) pares++;
24         else impares++;
25     }
26 }
27
28 int main() {
29     vector<int> numeros = {2,5,6,7,8};
30     int pares, impares;
31
32     int suma = calcularSuma(numeros);
33     double promedio = calcularPromedio(numeros);
34     contarParesImpares(numeros, pares, impares);
35
36     cout << "Suma: " << suma << endl;
37     cout << "Promedio: " << promedio << endl;
38     cout << "Pares: " << pares << ", Impares: " << impares << endl;
39
40     return 0;
41 }
42

```

Suma: 28
Promedio: 5.6
Pares: 3, Impares: 2

Process exited after 0.8151 seconds with return value 0
Presione una tecla para continuar . . .

• EJERCICIO 40: funciones + paso por valor + array

Enunciado:

Cree un programa en C++ que utilice un arreglo de números enteros y funciones con paso por valor. El programa debe enviar el arreglo a las funciones para calcular el valor mayor, el valor menor y la cantidad de números positivos, negativos y ceros, sin alterar los datos originales.

```

4 //No altere los datos originales.
5 //No altere los datos originales.
6 #include <iostream>
7 using namespace std;
8
9 int calcularMayor(int arr[], int n) {
10     int mayor = arr[0];
11     for (int i = 1; i < n; i++)
12         if (arr[i] > mayor) mayor = arr[i];
13     return mayor;
14 }
15
16 int calcularMenor(int arr[], int n) {
17     int menor = arr[0];
18     for (int i = 1; i < n; i++)
19         if (arr[i] < menor) menor = arr[i];
20     return menor;
21 }
22
23 void contarPosNegCeros(int arr[], int n, int &pos, int &neg, int &ceros) {
24     pos = neg = ceros = 0;
25     for (int i = 0; i < n; i++) {
26         if (arr[i] > 0) pos++;
27         else if (arr[i] < 0) neg++;
28         else ceros++;
29     }
30 }
31
32 int main() {
33     int arr[] = {2,-3,0,5,-1};
34     int pos, neg, ceros;
35
36     int mayor = calcularMayor(arr,5);
37     int menor = calcularMenor(arr,5);
38     contarPosNegCeros(arr,5,pos,neg,ceros);
39
40     cout << "Mayor: " << mayor << ", Menor: " << menor << endl;
41     cout << "Positivos: " << pos << ", Negativos: " << neg << ", Ceros: " << ceros << endl;
42
43     return 0;
44 }
45

```

Mayor: 5, Menor: -3
Positivos: 2, Negativos: 2, Ceros: 1

Process exited after 0.8277 seconds with return value 0
Presione una tecla para continuar . . .

• EJERCICIO 41: EJERCICIO INTEGRADOR

Enunciado:

Desarrolle un programa en C++ que permita gestionar las ventas diarias de una tienda.

El programa 41.1 CALIFICACIONES DE UN ESTUDIANTE

Enunciado:

Desarrolle un programa en C++ que permita gestionar las **calificaciones de un estudiante**.

El programa debe cumplir con los siguientes requisitos:

- Usar un **vector de números reales** para almacenar las calificaciones.
- Utilizar **funciones** para ingresar las calificaciones y realizar los cálculos.
- Guardar las calificaciones en un **archivo de texto**.
- Leer los datos desde el archivo.
- Calcular el **total de calificaciones** y el **promedio**.

- Mostrar cuántas calificaciones están **por encima del promedio** y cuántas están **por debajo del promedio**.
- Evitar repetir cálculos innecesarios.

```

3 //Desarrolle un programa en C++ que permita gestionar las calificaciones de un estudiante.
4
5 #include <iostream>
6 #include <vector>
7 #include <fstream>
8 using namespace std;
9
10 void ingresarCalificaciones(vector<double> &calificaciones) {
11     for (int i = 0; i < calificaciones.size(); i++) {
12         cout << "Ingrese calificación " << i+1 << ": ";
13         cin >> calificaciones[i];
14     }
15 }
16
17 void guardarArchivo(const vector<double> &calificaciones, string nombreArchivo) {
18     ofstream archivo(nombreArchivo);
19     for (int i = 0; i < calificaciones.size(); i++)
20         archivo << calificaciones[i] << endl;
21     archivo.close();
22 }
23
24 vector<double> leerArchivo(string nombreArchivo) {
25     vector<double> calificaciones;
26     ifstream archivo(nombreArchivo);
27     double nota;
28     while (archivo >> nota) calificaciones.push_back(nota);
29     archivo.close();
30     return calificaciones;
31 }
32
33 void calcularTotalPromedio(const vector<double> &calificaciones, double &total, double &promedio) {
34     total = 0;
35     for (double nota : calificaciones) total += nota;
36     promedio = total / calificaciones.size();
37 }
38
39 int contarArribaDeBajo(const vector<double> &calificaciones, double promedio, bool arriba) {

```

```

C:\Users\USUARIO\OneDrive\
Ingrese calificación 1: 9
Ingrese calificación 2: 8
Ingrese calificación 3: 7
Ingrese calificación 4: 8
Ingrese calificación 5: 9
Total de calificaciones: 41
Promedio: 8.2
Cantidad arriba del promedio: 2
Cantidad debajo del promedio: 3

-----
Process exited after 7.488 seconds with return value 0
Presione una tecla para continuar . . .

```

El programa 41.2: VENTAS

El programa debe desarrollarse en C++ y compilar correctamente.

Se debe utilizar un **vector<double>** para ingresar y almacenar inicialmente los valores de las ventas.

El ingreso de las ventas debe realizarse mediante una **función** que reciba el vector por referencia.

Una vez ingresados los datos, las ventas deben **guardarse en un archivo de texto** llamado ventas.txt.

El programa debe **leer las ventas desde el archivo** y almacenar esos valores en un **nuevo vector<double>**.

Todos los **cálculos** deben realizarse **únicamente** con el vector cargado desde el archivo.

Se deben usar funciones para:

- Calcular el **total de ventas**.
- Calcular el **promedio de ventas** (este cálculo debe hacerse una sola vez).

Utilizando el promedio calculado, el programa debe determinar:

- Cuántas ventas están **por encima del promedio**.
- Cuántas ventas están **por debajo del promedio**.

El programa debe mostrar en pantalla:

- Total de ventas.
- Promedio de ventas.
- Cantidad de ventas mayores y menores al promedio.

The screenshot shows a C++ program in a code editor on the left and its execution output in a terminal window on the right. The code defines functions for calculating the total and average of sales, and for counting sales above and below the average. The main function reads 5 sales values from the user, calculates the total and average, and then counts how many sales are above and below the average. The output window shows the user input for 5 sales (23, 45, 67, 54, 98), the calculated total (287), average (57.4), and counts (2 above, 3 below the average).

```

27     archivo.close();
28     return ventas;
29 }
30
31 void calcularTotalPromedio(const vector<double> &ventas, double &total, double &promedio) {
32     total = 0;
33     for (double v : ventas) total += v;
34     promedio = total / ventas.size();
35 }
36
37 int contarArribaDebajo(const vector<double> &ventas, double promedio, bool arriba) {
38     int count = 0;
39     for (double v : ventas) {
40         if (arriba && v > promedio) count++;
41         else if (!arriba && v < promedio) count++;
42     }
43     return count;
44 }
45
46 int main() {
47     const int N = 5;
48     vector<double> ventas(N);
49     ingresarVentas(ventas);
50     guardarArchivo(ventas, "ventas.txt");
51     vector<double> ventasArchivo = leerArchivo("ventas.txt");
52
53     double total, promedio;
54     calcularTotalPromedio(ventasArchivo, total, promedio);
55     int arriba = contarArribaDebajo(ventasArchivo, promedio, true);
56     int debajo = contarArribaDebajo(ventasArchivo, promedio, false);
57
58     cout << "Total de ventas: " << total << endl;
59     cout << "Promedio de ventas: " << promedio << endl;
60     cout << "Cantidad de ventas mayores al promedio: " << arriba << endl;
61     cout << "Cantidad de ventas menores al promedio: " << debajo << endl;
62
63     return 0;
64 }
65

```

```

C:\Users\USUARIO\OneDrive\
Ingreso venta 1: 23
Ingreso venta 2: 45
Ingreso venta 3: 67
Ingreso venta 4: 54
Ingreso venta 5: 98
Total de ventas: 287
Promedio de ventas: 57.4
Cantidad de ventas mayores al promedio: 2
Cantidad de ventas menores al promedio: 3

-----
Process exited after 8.112 seconds with return value 0
Presione una tecla para continuar . . . |

```

No se permite repetir cálculos innecesarios ni recalcular el promedio dentro de ciclos.

No se permite el uso de variables globales.

El código debe estar correctamente indentado y comentado.

Conclusiones

Durante la realización del taller se logró:

- Comprender y aplicar estructuras de datos lineales (arreglos y vectores) y bidimensionales (matrices).
- Implementar estructuras (`struct`) y trabajar con archivos para almacenar y recuperar información.
- Desarrollar programas modulares usando funciones, evitando redundancia y errores de cálculo.
- Aplicar condiciones y ciclos para resolver problemas estadísticos y lógicos.

Dificultades encontradas:

- Al principio hubo confusión en el manejo de vectores y arrays, especialmente al calcular máximos, mínimos y promedios.
- La lectura y escritura de archivos requirió cuidado para evitar errores de formato y pérdida de datos.
- La organización de funciones y parámetros por referencia necesitó planificación para evitar modificar valores originales por error.

Repositorio GitHub

ENLACE

https://github.com/Karen-Lozano/Taller_Cpp_Estructuras_Funciones_Karen_Lozano/tree/413bd304c0647608bafa0662fa303d1f20d698a8