



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:

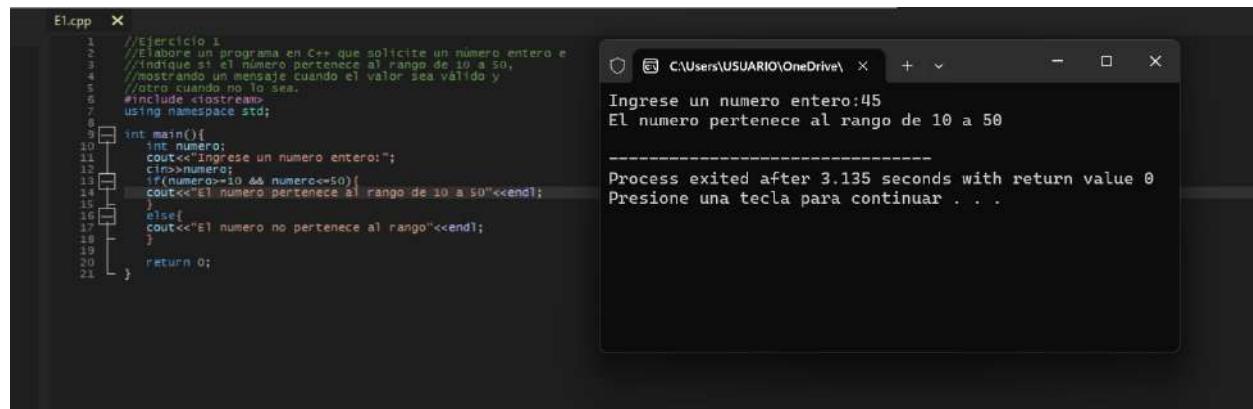
- Gestionar datos en estructuras lineales y bidimensionales.
- Aplicar condiciones y ciclos para cálculos específicos.
- Manejar archivos para almacenar y recuperar información.
- Implementar funciones para modularizar el código y evitar redundancia.

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 code editor window for 'Ejercicio 1' and a terminal window for its execution.

Code Editor (Ejercicio 1):

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

Terminal Output:

```
Ingrese un numero entero:45
El numero 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.

The screenshot shows a code editor with two tabs: E1.cpp and E2.cpp. The E2.cpp tab is active, displaying the following C++ code:

```

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
11 int main(){
12     int numero;
13     cout<<"Ingrese un numero entero:";
14     cin>>numero;
15
16     if(numero<10 || numero>50){
17         cout<<"El numero no se encuentra en el rango de 10 a 50."<<endl;
18     }
19     else{
120         cout<<"El numero pertenece al rango de 10 a 50."<<endl;
111    }
112
113    return 0;
114 }

```

To the right of the code editor is a terminal window titled 'C:\Users\USUARIO\OneDrive\'. It displays the following output:

```

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.

The screenshot shows a code editor with three tabs: E1.cpp, E2.cpp, and E3.cpp. The E3.cpp tab is active, displaying the following C++ code:

```

1 //Ejercicio 3
2 //Desarrolle un programa que muestre en pantalla los numeros
3 //del 1 al 10, pero indicando al final cuantos numeros 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
19     return 0;
20 }

```

To the right of the code editor is a terminal window titled 'C:\Users\USUARIO\OneDrive\'. It displays the following output:

```

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.

The screenshot shows a code editor with four tabs: E1.cpp, E2.cpp, E3.cpp, and E4.cpp. The E4.cpp tab is active, displaying the following C++ code:

```

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

```

To the right of the code editor is a terminal window titled 'C:\Users\USUARIO\OneDrive\'. It displays the following output:

```

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.

```

E5.cpp x E2.cpp x E3.cpp x E4.cpp x E5.cpp x
1 //Ejercicio 5
2 //Elabore un programa que solicite un numero entero positivo y
3 //muestre su tabla de multiplicar del 1 al 10,
4 //ademas de calcular la suma de los resultados obtenidos.
5
6 #include <iostream>
7 using namespace std;
8
9 int main(){
10     int num;
11     cout<<"Ingrese un numero entero positivo:"; 
12     cin>>num;
13     if(num>0){
14         cout<<"TABLA DE MULTIPLICAR"<<endl;
15         for(int i=1; i<=10; i++){
16             cout<<num<<'x'<<i<<"="<<num*i<<endl;
17             suma+=num*i;
18         }
19         cout<<"La suma de sus resultados son:"<<suma<<endl;
20     }
21     else{
22         cout<<"El numero ingresado no es valido. Intentelo de nuevo!"<<endl;
23     }
24
25     return 0;
26 }

```

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.

```

E6.cpp x E2.cpp x E3.cpp x E4.cpp x E5.cpp x E6.cpp x
1 //Ejercicio 6
2 //Desarrolle un programa que muestre los numeros del 1 al 20
3 //y que para cada numero si es par o impar,
4 //utilizando estructuras for e if.
5 #include <iostream>
6 using namespace std;
7
8 int main(){
9     for(int i=1; i<=20; i++){
10         if(i%2==0){
11             cout<<i<<" es par"<<endl;
12         }
13         else{
14             cout<<i<<" es impar"<<endl;
15         }
16     }
17
18     return 0;
19 }

```

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.

```

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

```

Los numeros multiplos de 3 son:16
Los numeros multiplos 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.

```

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

```

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 . . . |

- Command: g++-exe "C:\Users\USUARIO\OneDrive\Documentos\E8.cpp" -o "C:\Users\USUARIO\OneDrive\Documentos\E8.exe" -Wall -Wextra -g3 -finput-charset=utf-8 -fexec-charset=windows-1252 -I"C:\Program Files (x86)\Dev-Cpp\MinGW64\incl"

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.

The screenshot shows a code editor with multiple tabs (E1.cpp, E2.cpp, E3.cpp, E4.cpp, E5.cpp, E6.cpp, E7.cpp, E8.cpp, E9.cpp, E10.cpp) and a terminal window. The terminal window displays the following 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 . . .
```

Below the terminal, the command used for compilation is shown: g++ -o "C:\Users\USUARIO\OneDrive\Documentos\E10.cpp" "C:\Users\USUARIO\OneDrive\Documentos\E10.cpp". Compilation results show 0 errors and 0 warnings.

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.

The screenshot shows a code editor with multiple tabs (E1.cpp, E2.cpp, E3.cpp, E4.cpp, E5.cpp, E6.cpp, E7.cpp, E8.cpp, E9.cpp) and a terminal window. The terminal window displays the following 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

The screenshot shows a code editor with a file named E1.cpp. The code is a solution for Exercise 11, which asks to calculate the sum, average, and percentages of positive and negative numbers from an array of 10 floats. The code uses nested loops to iterate through the array and conditional statements to count positives and negatives. It then prints the total sum, average, and percentages. The terminal window shows the input of 10 numbers (1, 4, 2, 5, 3, 6, 4, 7, 5, 8) and the output of the program's results.

```

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 de los valores
5 // El porcentaje de números positivos
6 // El porcentaje de números negativos
7
8 #include <iostream>
9 using namespace std;
10
11 int main(){
12
13     float numeros[10];
14     float suma=0;
15     float positivos=0, negativos=0;
16
17     cout<<"Ingrese 10 numeros: "<<endl;
18     for(int i=0; i<10; i++){
19         cout<<i+1<<" ";
20         cin>>numeros[i];
21         suma+=numeros[i];
22
23         if(numeros[i]>0){
24             positivos++;
25         }
26         else if(numeros[i]<0){
27             negativos++;
28         }
29     }
30
31     cout<<"La suma total de los valores es: "<<suma<<endl;
32     cout<<"El promedio es: "<<suma/10<<endl;
33     cout<<"Porcentaje positivos: "<<(positivos/10)*100<<endl;
34     cout<<"Porcentaje negativos: "<<(negativos/10)*100<<endl;
35
36     return 0;
37 }
```

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

The screenshot shows a code editor with a file named E12.cpp. The code is a solution for Exercise 12, which asks to find the maximum, minimum, and difference between them in an array of 8 integers. It uses a loop to iterate through the array and compare each element with the current maximum and minimum. The terminal window shows the input of 8 numbers (2, 3, 4, 5, 6, 7, 8, 9) and the output of the program's results.

```

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

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
Número mayor: 9
Número 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 //Desarrolla 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 }
36

```

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 // Cuantos estudiantes aprobaron (≥ 7)
5 // Cuantos reprobados (< 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 calificacion " << i + 1 << " (0-10): ";
19             cin >> calificaciones[i];
20         } while (calificaciones[i] < 0 || calificaciones[i] > 10);
21
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

```

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

The screenshot shows a code editor with multiple tabs (E1.cpp, E2.cpp, etc.) and a terminal window. The terminal displays the execution of a program that calculates the average of 15 numbers entered by the user. The program also counts positive and negative numbers.

```

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        if (countPos > 0)
26            cout << "Promedio de positivos: " << sumaPos / countPos << endl;
27        else
28            cout << "No hay numeros positivos." << endl;
29
30        if (countNeg > 0)
31            cout << "Promedio de negativos: " << sumaNeg / countNeg << endl;
32        else
33            cout << "No hay numeros negativos." << endl;
34
35    }
36
37    return 0;
38 }
```

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

The screenshot shows a code editor with multiple tabs (E1.cpp, E2.cpp, etc.) and a terminal window. The terminal displays the execution of a program that calculates the sum and average of 10 numbers entered by the user, and counts how many numbers are above the average.

```

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 //Desarrolle un programa con vectores que permita ingresar 8 valores y calcule:
3 // El valor maximo
4 // El valor minimo
5 // El rango (maximo - minimo)
6 #include <iostream>
7 using namespace std;
8
9 int main() {
10     int valores[8];
11     int maximo, minimo;
12
13     for (int i = 0; i < 8; i++) {
14         cout << "Ingrese el valor " << i + 1 << ": ";
15         cin >> valores[i];
16     }
17
18     maximo = minimo = valores[0];
19
20     for (int i = 1; i < 8; i++) {
21         if (valores[i] > maximo) maximo = valores[i];
22         if (valores[i] < minimo) minimo = valores[i];
23     }
24
25     cout << "Valor maximo: " << maximo << endl;
26     cout << "Valor minimo: " << minimo << endl;
27     cout << "Rango: " << maximo - minimo << endl;
28
29     return 0;
30 }

```

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 sumPos = 0, sumNeg = 0, sumAbs = 0;
13
14     for (int i = 0; i < 12; i++) {
15         cout << "Ingrese el valor " << i + 1 << ": ";
16         cin >> valores[i];
17
18         if (valores[i] > 0) sumPos += valores[i];
19         else if (valores[i] < 0) sumNeg += valores[i];
20
21         sumAbs += abs(valores[i]);
22     }
23
24     cout << "Suma de positivos: " << sumPos << endl;
25     cout << "Suma de negativos: " << sumNeg << endl;
26     cout << "Suma del valor absoluto total: " << sumAbs << endl;
27
28     return 0;
29 }

```

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: 0
Ingrese el valor 10: 1
Ingrese el valor 11: 2
Ingrese el valor 12: 3
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

The screenshot shows a code editor with multiple tabs (E1.cpp, E2.cpp, E3.cpp, E4.cpp, E5.cpp, E6.cpp, E7.cpp, E8.cpp, E9.cpp, E10.cpp, E11.cpp, E12.cpp, E13.cpp, E14.cpp, E15.cpp) and a terminal window. The terminal window displays the following 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

The screenshot shows a code editor with multiple tabs (E15.cpp, E16.cpp, E17.cpp, E18.cpp, E19.cpp, E20.cpp) and a terminal window. The terminal window displays the following 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 . . .
```

At the bottom of the terminal window, there is additional information about the process:

```
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

```

E1.cpp * E2.cpp * E3.cpp * E4.cpp * E5.cpp * E6.cpp * E7.cpp * E8.cpp * E9.cpp * E10.cpp * E11.cpp * E12.cpp * E13.cpp * E14.cpp * E15.cpp * E16.cpp * E17.cpp * E18.cpp * E19.cpp * E20.cpp * E21.cpp * E22.cpp
  
```

Ejercicio 21

```

1 //Ejercicio 21
2 //Elabore un programa que permita ingresar 10 números en un arreglo y determine
3 //cuantos están dentro en el rango de 20 a 80
4 //Cuantos están fuera de ese rango
5 //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
18         if (numeros[i] >= 20 && numeros[i] <= 80) dentro++;
19         else fuera++;
20     }
21
22     cout << "Cantidad dentro del rango 20-80: " << dentro << endl;
23     cout << "Cantidad fuera del rango: " << fuera << endl;
24     cout << "Porcentaje dentro del rango: " << (dental * 100.0 / 10) << "%" << endl;
25     cout << "Porcentaje fuera del rango: " << (fuerza * 100.0 / 10) << "%" << endl;
26
27 }
28 return 0;
  
```

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 0
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

```

E1.cpp * E2.cpp * E3.cpp * E4.cpp * E5.cpp * E6.cpp * E7.cpp * E8.cpp * E9.cpp * E10.cpp * E11.cpp * E12.cpp * E13.cpp * E14.cpp * E15.cpp * E16.cpp * E17.cpp * E18.cpp * E19.cpp * E20.cpp * E21.cpp * E22.cpp
  
```

Ejercicio 22

```

1 //Ejercicio 22
2 //Desarrolle un programa que permita ingresar 15 valores en un vector y calcule:
3 //Cuantos valores son multiplos de 3
4 //Cuantos valores son multiplos de 5
5 //Cuantos valores son multiplos 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
18         if (numeros[i] % 3 == 0) mult3++;
19         if (numeros[i] % 5 == 0) mult5++;
20         if (numeros[i] % 3 == 0 && numeros[i] % 5 == 0) multAmbos++;
21
22     }
23
24     cout << "Multiplos de 3: " << mult3 << endl;
25     cout << "Multiplos de 5: " << mult5 << endl;
26     cout << "Multiplos de ambos: " << multAmbos << endl;
27
28 }
29 return 0;
  
```

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
Multiplos de 3: 7
Multiplos de 5: 3
Multiplos de ambos: 1

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

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\USUARIO\OneDrive\Documentos\E22.exe
- Output Size: 13.800185203552 MiB
- Compilation Time: 1.95s.

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 // Desarrollar 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 // Desarrollar 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
22         suma += notas[i];
23     }
24
25     mayor = menor = notas[0];
26
27     for (int i = 0; i < 10; i++) {
28         if (notas[i] >= 7) aprobados++;
29         else reprobados++;
30
31         if (notas[i] > mayor) mayor = notas[i];
32         if (notas[i] < menor) menor = notas[i];
33     }
34
35     double promedio = suma / 10;
36     double porcAprob = aprobados * 10.0 // 10% por cada estudiante
37     double porcReprob = reprobados * 10.0;
38
39     cout << "\n--- RESUMEN FINAL ---" << endl;
40     cout << "Promedio general: " << promedio << endl;
41     cout << "Nota mayor: " << mayor << endl;
42     cout << "Nota menor: " << menor << endl;
43     cout << "Porcentaje de aprobados: " << porcAprob << "%" << endl;
44     cout << "Porcentaje de Reprobados: " << porcReprob << "%" << endl;
45
46     return 0;
47 }

```

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

```

4 //1. Mostrar la suma y la resta
5 //2. Mostrar el mayor de los dos numeros
6 //3. Salir
7 #include <iostream>
8 using namespace std;
9
10 int main() {
11     int opcion;
12     double num1 = 0, num2 = 0;
13
14     do {
15         cout << "\n--- MENU ---\n";
16         cout << "1. Ingresar dos numeros\n";
17         cout << "2. Mostrar suma y resta\n";
18         cout << "3. Mostrar el mayor de los dos numeros\n";
19         cout << "4. Salir\n";
20         cout << "Ingrese una opcion: ";
21         cin >> opcion;
22
23         switch(opcion) {
24             case 1:
25                 cout << "Ingrese el primer numero: ";
26                 cin >> num1;
27                 cout << "Ingrese el segundo numero: ";
28                 cin >> num2;
29                 break;
30             case 2:
31                 cout << "Suma: " << num1 + num2 << endl;
32                 cout << "Resta: " << num1 - num2 << endl;
33                 break;
34             case 3:
35                 if (num1 > num2) cout << "El mayor es: " << num1 << endl;
36                 else if (num2 > num1) cout << "El mayor es: " << num2 << endl;
37                 else cout << "Ambos numeros son iguales." << endl;
38                 break;
39             case 4:
40                 cout << "Saliendo del programa..." << endl;
41                 break;
42             default:
43                 cout << "Opcion no valida." << endl;
44         }
45     } while(opcion != 4);
46
47     return 0;
48 }

```

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\USUARIO\OneDrive\Documentos\E26.exe
- Output Size: 13.8001852035522 MB

--- 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

The screenshot shows a C++ development environment with several tabs open. The active tab contains the following code:

```

1 //1. Verificar si el numero 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 numero\n";
13        cout << "2. Verificar si el numero esta en el rango 1-100\n";
14        cout << "3. Verificar si el numero es par o impar\n";
15        cout << "4. Salir\n";
16        cout << "Ingresue una opcion: ";
17        cin >> opcion;
18
19        switch(opcion) {
20            case 1:
21                cout << "Ingresue el numero: ";
22                cin >> numero;
23                break;
24            case 2:
25                if (numero >= 1 && numero <= 100)
26                    cout << "El numero esta en el rango 1-100.\n" << endl;
27                else
28                    cout << "El numero esta fuera del rango.\n" << endl;
29                break;
30            case 3:
31                if (numero % 2 == 0) cout << "El numero es par.\n" << endl;
32                else cout << "El numero es impar.\n" << endl;
33                break;
34            case 4:
35                cout << "Saliendo del programa...\n" << endl;
36                break;
37            default:
38                cout << "Opcion no valida.\n" << endl;
39        }
40    } while(opcion != 4);
41
42    return 0;
43}

```

Output window:

```

--- MENU ---
1. Ingresar un numero
2. Verificar si el numero esta en el rango 1-100
3. Verificar si el numero es par o impar
4. Salir
Ingruese una opcion: 1
Ingruese el numero: 23

--- MENU ---
1. Ingresar un numero
2. Verificar si el numero esta en el rango 1-100
3. Verificar si el numero es par o impar
4. Salir
Ingruese una opcion: 2
El numero esta en el rango 1-100.

--- MENU ---
1. Ingresar un numero
2. Verificar si el numero esta en el rango 1-100
3. Verificar si el numero es par o impar
4. Salir
Ingruese una opcion: 3
El numero es impar.

--- MENU ---
1. Ingresar un numero
2. Verificar si el numero esta en el rango 1-100

```

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

The screenshot shows a C++ development environment with several tabs open. The active tab contains the following code:

```

1 //1. Ingresar 10 numeros en un arreglo
2 //2. Mostrar el promedio
3 //3. Mostrar el mayor y el menor
4 //4. Contar numeros positivos y negativos
5 //5. Salir
6
7 #include <iostream>
8 using namespace std;
9
10 int main() {
11     int opcion;
12     int numeros[10];
13     bool ingresado = false;
14
15     do {
16         cout << "\n--- MENU ---\n";
17         cout << "1. Ingresar 10 numeros en un arreglo\n";
18         cout << "2. Mostrar el promedio\n";
19         cout << "3. Mostrar el mayor y el menor\n";
20         cout << "4. Contar numeros positivos y negativos\n";
21         cout << "5. Salir\n";
22         cout << "Ingresue una opcion: ";
23         cin >> opcion;
24
25         switch(opcion) {
26             case 1:
27                 for (int i = 0; i < 10; i++) {
28                     cout << "Ingruese el numero " << i + 1 << ": ";
29                     cin >> numeros[i];
30                 }
31                 ingresado = true;
32                 break;
33             case 2:
34                 if (ingresado) cout << "Primero ingrese los numeros.\n" << endl;
35                 else {
36                     double sum = 0;
37                     for (int i = 0; i < 10; i++) sum += numeros[i];
38                     cout << "Promedio: " << sum / 10 << endl;
39                 }
40                 break;
41             case 3:
42                 if (ingresado) cout << "Primero ingrese los numeros.\n" << endl;
43                 else {
44                     int maximo = numeros[0], minimo = numeros[0];
45                     for (int i = 1; i < 10; i++) {
46                         if (numeros[i] > maximo) maximo = numeros[i];
47                         if (numeros[i] < minimo) minimo = numeros[i];
48                     }
49                     cout << "Mayor: " << maximo << ", Menor: " << minimo << endl;
50                 }
51                 break;
52             case 4:
53                 if (ingresado) cout << "Primero ingrese los numeros.\n" << endl;
54                 else {
55                     int positivos = 0, negativos = 0;
56                     for (int i = 0; i < 10; i++) {
57                         if (numeros[i] > 0) positivos++;
58                         if (numeros[i] < 0) negativos++;
59                     }
60                     cout << "Positivos: " << positivos << ", Negativos: " << negativos << endl;
61                 }
62                 break;
63             default:
64                 cout << "Opcion no valida.\n" << endl;
65         }
66     } while(opcion != 5);
67
68    return 0;
69}

```

Output window:

```

--- MENU ---
1. Ingresar 10 numeros en un arreglo
2. Mostrar el promedio
3. Mostrar el mayor y el menor
4. Contar numeros positivos y negativos
5. Salir
Ingresue una opcion: 1
Ingruese el numero 1: 2
Ingruese el numero 2: 3
Ingruese el numero 3: 4
Ingruese el numero 4: 5
Ingruese el numero 5: 6
Ingruese el numero 6: 7
Ingruese el numero 7: 8
Ingruese el numero 8: 9
Ingruese el numero 9: 0
Ingruese el numero 10: 9

--- MENU ---
1. Ingresar 10 numeros en un arreglo
2. Mostrar el promedio
3. Mostrar el mayor y el menor
4. Contar numeros positivos y negativos
5. Salir
Ingresue una opcion: 2
Promedio: 5.3

--- MENU ---
1. Ingresar 10 numeros en un arreglo

```

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

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

```

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++) {
22            cout << "Suma de fila " << i+1 << ":" << sumafila << endl;
23            sumafila += matriz[i][j];
24        }
25
26        cout << "Suma total de la matriz: " << sumaTotal << endl;
27    }
28
29    return 0;
30 }

```

The terminal window shows the program's output:

```

Ingresar el valor de fila 1 columna 1: 2
Ingresar el valor de fila 1 columna 2: 3
Ingresar el valor de fila 1 columna 3: 4
Ingresar el valor de fila 2 columna 1: 5
Ingresar el valor de fila 2 columna 2: 6
Ingresar el valor de fila 2 columna 3: 7
Ingresar el valor de fila 3 columna 1: 8
Ingresar el valor de fila 3 columna 2: 9
Ingresar 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

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

```

1 //Ejercicio 30
2 //Elabore un programa que permita ingresar una matriz 2x4 y determine:
3 //El numero mayor
4 //La posicion (fila y columna) del numero 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        cout << "Numero mayor: " << mayor << " en fila " << filaMayor+1 << " columna " << colMayor+1 << endl;
33    }
34
35    return 0;
36 }

```

The terminal window shows the program's output:

```

Ingresar el valor de fila 1 columna 1: 2
Ingresar el valor de fila 1 columna 2: 3
Ingresar el valor de fila 1 columna 3: 4
Ingresar el valor de fila 1 columna 4: 5
Ingresar el valor de fila 2 columna 1: 6
Ingresar el valor de fila 2 columna 2: 7
Ingresar el valor de fila 2 columna 3: 8
Ingresar el valor de fila 2 columna 4: 9
Número 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

The screenshot shows a code editor with a C++ file named `Ejercicio_31.cpp`. The code reads a 4x4 matrix from the user and counts the number of positive, negative, and zero values. The output window shows the input values for each cell of the matrix, followed by the counts: 16 positives, 8 negatives, and 0 zeros.

```

1 //Ejercicio_31
2 //Realice un programa que permita ingresar una matriz 4x4 y cuente cuantos 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 }

```

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: 4
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.

The screenshot shows a code editor with a C++ file named `EJERCICIO_32_struct + vector.cpp`. The code defines a `Cliente` struct with `nombre` and `edad` fields. It then creates a vector of `Cliente` objects and adds two clients to it. The output window shows the input for two clients and the resulting list of registered clients.

```

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 //que permita ingresar varios clientes usando un vector y luego mostrar en pantalla la informacion 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 << "Cuantos 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 }

```

Cuantos 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 //Realice 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 edad.
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 >= 18)
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 
```

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;
39     }
40     archivo.close();
41
42     // Leer desde archivo
43     vector<Cliente> clientesArchivo;
44     ifstream archivoLeer("clientes.txt");
45     string nombre;
46     while (getline(archivoLeer, nombre)) {
47
48

```

Cuantos clientes desea ingresar? 2
Nombre del cliente 1: KAREN
Edad del cliente 1: 21
Consumo mensual: 100
Nombre del cliente 2: LUIS
Edad del cliente 2: 20
Consumo mensual: 200
Total de consumo: 300
Promedio de consumo: 150
Cliente con mayor consumo: LUIS (200)

Process exited after 21.79 seconds with return value 0
Presione una tecla para continuar . . . |

- 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 reprende (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 reprende (promedio mayor o igual a 7).
6 #include <iostream>
7 #include <vector>
8 #include <fstream>
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 << " "
36         << estudiantes[i].nota2 << " " << estudiantes[i].nota3 << endl;
37     }
38     archivo.close();
39
40     // Leer desde archivo
41     vector<Estudiante> lista;
42     ifstream archivoLeer("estudiantes.txt");
43     string nombre;
44     while (getline(archivoLeer, nombre)) {
45
46

```

Cuantos estudiantes desea ingresar? 2
Nombre del estudiante 1: KAREN
Nota 1: 9
Nota 2: 8
Nota 3: 10
Nombre del estudiante 2: LUIS
Nota 1: 8
Nota 2: 9
Nota 3: 7
Ingrese el nombre del estudiante a buscar: LUIS
Notas: 8, 9, 7
Promedio: 8
Estado: Aprobado

Process exited after 24.71 seconds with return value 0
Presione una tecla para continuar . . . |

- 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.

The screenshot shows a code editor with a C++ file named 'Ejercicio38.cpp'. The code includes functions to fill a vector with integers, calculate the sum and average, and find the maximum and minimum values. Below the editor is a terminal window titled 'C:\Users\USUARIO\OneDrive\'. It displays the program's output: five integer inputs (4, 5, 6, 7, 8), their sum (30), average (6), and the maximum (8) and minimum (4). The terminal also shows the exit message 'Process exited after 8.653 seconds with return value 0'.

```

Ejercicio38.cpp
4 //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.
5
6 #include <iostream>
7 #include <vector>
8 using namespace std;
9
10 // Función para llenar el vector
11 void llenarVector(vector<int>& v) {
12     for (int i = 0; i < v.size(); i++) {
13         cout << "Ingrese el numero " << i+1 << ": ";
14         cin >> v[i];
15     }
16 }
17
18 // Función para calcular suma y promedio
19 void calcularSumaPromedio(const vector<int>& v, int &suma, double &promedio) {
20     suma = 0;
21     for (int i = 0; i < v.size(); i++) suma += v[i];
22     promedio = suma / double(v.size());
23 }
24
25 // Función para calcular mayor y menor
26 void calcularMayorMenor(const vector<int>& v, int &mayor, int &menor) {
27     mayor = menor = v[0];
28     for (int i = 1; i < v.size(); i++) {
29         if (v[i] > mayor) mayor = v[i];
30         if (v[i] < menor) menor = v[i];
31     }
32 }
33
34 int main() {
35     vector<int> numeros; // Cambiar tamaño si se desea
36     int suma, mayor, menor;
37     double promedio;
38
39     llenarVector(numeros);
40     calcularSumaPromedio(numeros, suma, promedio);
41     calcularMayorMenor(numeros, mayor, menor);
42
43     cout << "Suma: " << suma << endl;
44     cout << "Promedio: " << promedio << endl;
45     cout << "Mayor: " << mayor << endl;
46     cout << "Menor: " << menor << endl;
}

```

- 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.

The screenshot shows a code editor with a C++ file named 'Ejercicio39.cpp'. The code includes functions to fill an array with double values, calculate the total and average, and count how many values are above the average. Below the editor is a terminal window titled 'C:\Users\USUARIO\OneDrive\'. It displays five double inputs (23, 34, 32, 54, 67), their total (210), average (42), and the count of values above the average (2). The terminal also shows the exit message 'Process exited after 9.256 seconds with return value 0'.

```

Ejercicio39.cpp
1 // EJERCICIO 39: funciones + paso por referencia + array
2
3 //Enunciado:
4 //Cree un programa en C++ que utilice un arreglo de números reales y funciones con paso por referencia.
5 //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.
6
7 #include <iostream>
8 using namespace std;
9
10 void llenarArreglo(double arr[], int n) {
11     for (int i = 0; i < n; i++) {
12         cout << "Ingrese la venta " << i+1 << ": ";
13         cin >> arr[i];
14     }
15 }
16
17 void calcularTotalPromedio(const double arr[], int n, double &total, double &promedio) {
18     total = 0;
19     for (int i = 0; i < n; i++) total += arr[i];
20     promedio = total / n;
21 }
22
23 int contarMayoresPromedio(const double arr[], int n, double promedio) {
24     int count = 0;
25     for (int i = 0; i < n; i++) {
26         if (arr[i] > promedio) count++;
27     }
28     return count;
29 }
30
31 int main() {
32     const int N = 5; // Cambiar tamaño si se desea
33     double ventas[N];
34     double total, promedio;
35
36     llenarArreglo(ventas, N);
37     calcularTotalPromedio(ventas, N, total, promedio);
38     int mayores = contarMayoresPromedio(ventas, N, promedio);
39
40     cout << "Total de ventas: " << total << endl;
41     cout << "Promedio de ventas: " << promedio << endl;
42     cout << "Ventas mayores al promedio: " << mayores << endl;
}

```

- 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
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

```

- 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 //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.
5
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[5] = {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 << endl;
41     cout << "Menor: " << menor << endl;
42 }

```

- 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.

The screenshot shows a code editor with a C++ file containing a program to manage student grades. The code includes functions for inputting grades into a vector, calculating the total and average, and counting grades above and below the average. To the right, a terminal window displays the execution of the program, asking for five grades (9, 2, 7, 8, 9), calculating a total of 41 and an average of 8.2, and then outputting the counts of grades above (3) and below (2) the average.

```

1 //desarrollar un programa en C++ que permita gestionar las calificaciones de un estudiante.
2
3 #include <iostream>
4 #include <vector>
5 #include <iomanip>
6 using namespace std;
7
8 void ingresarCalificaciones(vector<double> &calificaciones) {
9     for (int i = 0; i < calificaciones.size(); i++) {
10        cout << "Ingrese calificacion " << i+1 << ": ";
11        cin >> calificaciones[i];
12    }
13 }
14
15 void guardarArchivo(const vector<double> &calificaciones, string nombreArchivo) {
16     ofstream archivo(nombreArchivo);
17     for (int i = 0; i < calificaciones.size(); i++) {
18        archivo << calificaciones[i] << endl;
19    }
20    archivo.close();
21 }
22
23 vector<double> leerArchivo(string nombreArchivo) {
24     vector<double> calificaciones;
25     ifstream archivo(nombreArchivo);
26     double nota;
27     while (archivo >> nota) calificaciones.push_back(nota);
28     archivo.close();
29     return calificaciones;
30 }
31
32 void calcularTotalPromedio(const vector<double> &calificaciones, double &total, double &promedio) {
33     total = 0;
34     for (double nota : calificaciones) total += nota;
35     promedio = total / calificaciones.size();
36 }
37
38 int contarArribaDeabajo(const vector<double> &calificaciones, double promedio, bool arriba) {
39

```

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.

```

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

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/413bd304c0647608bafa06
62fa303d1f20d698a8](https://github.com/Karen-Lozano/Taller_Cpp_Estructuras_Funciones_Karen_Lozano/tree/413bd304c0647608bafa0662fa303d1f20d698a8)