



Nombre: Cristopher Cruz Ramos Jareth Alexander López Corona Nairda Tonantzin Morales del Real	
Carrera: TSU en Tecnologías de la Información, Área Desarrollo de Software Multiplataforma.	Grupo: 4ATI
Asignatura: Estructuras de Datos Aplicadas	Unidad: Arreglos
Profesor: MGTI Jorge Luis De La Paz Ramos	

1. Diseña un programa con un arreglo de 20 elementos llamado pares y guarda los 20 primeros números pares en dicho arreglo, muestra en pantalla el contenido del arreglo.

```
arreglos01.cpp  arreglo03.cpp  arreglo02.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int pares[20];
6      for ( int i = 0; i < 20; i++ ) {
7          pares[i] = i+1;
8          if ( pares[i] %2 == 0 ) {
9              cout << pares[i] << ", ";
10         }
11     }
12
13     return 0;
14 }
15
```

C:\Program Files\Ampss\www X + v

2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

Process exited after 0.08323 seconds with return value 0
Presione una tecla para continuar . . . |



2. programa que llene por teclado la nota de los alumnos de una clase y calcula la nota media del grupo. también muestra los alumnos con notas superiores a la media. el número de alumnos se lee por teclado este programa utiliza un arreglo de elementos que contendrá las notas de los alumnos. El tamaño del arreglo será el número de alumnos de la clase por lo tanto primero se pedirá por teclado el número de alumnos y a continuación se creará el arreglo. se realizan 3 recorridos sobre el arreglo, el primera para asignar a cada elemento las notas introducidas por el teclado el segundo para sumarlas y calcular la media, y el tercero para mostrar los alumnos con notas superiores.

```
arreglos01.cpp arreglos02.cpp arreglos03.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Ingrese cantidad de alumnos: ";
6     int tam;
7     cin >> tam;
8     int al[tam];
9     int suma = 0;
10
11     for ( int i = 0; i < tam; i++ ) {
12         cout << "Calificación de alumno "<< i << ": ";
13         cin >> al[i];
14         suma += al[i];
15     }
16
17     int prom = suma/tam;
18     cout << "Media de calificaciones: "<< prom << endl;
19     cout << "Calificaciones superiores a la media:" << endl;
20     for ( int i = 0; i < tam; i++ ) {
21         if ( al[i] > prom ) {
22             cout << "Calificación del alumno "<< i << ": ["<< al[i] << endl;
23         }
24     }
25
26     return 0;
27 }
28
```

```
C:\Program Files\Ampss\www X + v
Ingrese cantidad de alumnos: 15
Calificación de alumno 0: 10
Calificación de alumno 1: 9
Calificación de alumno 2: 8
Calificación de alumno 3: 9
Calificación de alumno 4: 10
Calificación de alumno 5: 7
Calificación de alumno 6: 6
Calificación de alumno 7: 8
Calificación de alumno 8: 9
Calificación de alumno 9: 10
Calificación de alumno 10: 8
Calificación de alumno 11: 8
Calificación de alumno 12: 9
Calificación de alumno 13: 7
Calificación de alumno 14: 6
Media de calificaciones: 8
Calificaciones superiores a la media:
Calificación del alumno 0: 10
Calificación del alumno 1: 9
Calificación del alumno 3: 9
Calificación del alumno 4: 10
Calificación del alumno 8: 9
Calificación del alumno 9: 10
Calificación del alumno 12: 9

Process exited after 21.14 seconds with return value 0
Presione una tecla para continuar . . . |
```

```
-----
- C++ Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe
- Command: g++.exe "C:\Program Files\Ampss\www\estructura_datos\C++\arreglos01.cpp"
Compilation results...
-----
```

3. Crear un arreglo que solicite 10 numeros y de ellos saque el digito mayor y muestre tambien su posicion dentro del arreglo. No se pueden introducir numeros repetidos.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int ar[10];
6
7     for ( int i = 0; i < 10; i++ ) {
8         cout << "Ingrese el número " << i << " del arreglo: ";
9         int num;
10        cin >> num;
11
12        bool existe = false;
13        for ( int ii = 0; ii < i; ii++ ) {
14            if ( ar[ii] == num ) {
15                existe = true;
16                break;
17            }
18        }
19
20        if ( existe ) {
21            cout << "Este número ya está en el arreglo, introduzca otro número:" << endl;
22            i--;
23        } else {
24            ar[i] = num;
25        }
26    }
27
28    int max = ar[0];
29    int pos = 0;
30
31    for ( int i = 1; i < 10; i++ ) {
32        if ( ar[i] > max ) {
33            max = ar[i];
34            pos = i;
35        }
36    }
37
38    cout << "El mayor número es: " << max << endl;
39    cout << "Su posición es: " << pos << endl;
40
41    return 0;
42 }
```

```
C:\Program Files\Ampss\www X + v
Ingrese el número 0 del arreglo: 15
Ingrese el número 1 del arreglo: 21
Ingrese el número 2 del arreglo: 30
Ingrese el número 3 del arreglo: 8
Ingrese el número 4 del arreglo: 17
Ingrese el número 5 del arreglo: 6
Ingrese el número 6 del arreglo: 6
Este número ya está en el arreglo, introduzca otro número:
Ingrese el número 6 del arreglo: 9
Ingrese el número 7 del arreglo: 11
Ingrese el número 8 del arreglo: 26
Ingrese el número 9 del arreglo: 28
Valor máximo = 30
En la posición: 2

Process exited after 27.6 seconds with return value 0
Presione una tecla para continuar . . . |
```

4. Realice un programa que calcule la media aritmética de N valores, además imprima los números ingresados en orden inverso.



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Cantidad de valores: ";
6     int n;
7     cin >> n;
8
9     int ar[n];
10    int suma = 0;
11
12    for ( int i = 0; i < n; i++ ) {
13        cout << "Ingrese el número " << i << " del arreglo: ";
14        cin >> ar[i];
15        suma += ar[i];
16    }
17    int prom = suma/n;
18
19    cout << "Cálculo de la media = " << prom << endl;
20    cout << "Orden inverso:" << endl;
21    for(int i = n - 1; i >= 0; i--) {
22        cout << ar[i] << " ";
23    }
24
25    return 0;
26 }
```

```
C:\Program Files\Ampss\www  X  +  v
Cantidad de valores: 6
Ingrese el número 0 del arreglo: 12
Ingrese el número 1 del arreglo: 14
Ingrese el número 2 del arreglo: 8
Ingrese el número 3 del arreglo: 11
Ingrese el número 4 del arreglo: 6
Ingrese el número 5 del arreglo: 5
Cálculo de la media = 9
Orden inverso:
5 6 11 8 14 12
-----
Process exited after 12.37 seconds with return value 0
Presione una tecla para continuar . . . |
```

5. Programa que lea el nombre y el sueldo de 20 empleados y muestre el nombre y el sueldo del empleado que más gana. Si hay mas de un empleado que mostrar en el resultado, debe mostrar todos los que más ganan.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 20;
6     string emp[n];
7     int sueldo[n];
8
9     for ( int i = 0; i < n; i++ ) {
10        cout << "Nombre del empleado "<< i <<": ";
11        cin >> emp[i];
12        cout << "Sueldo que tiene: ";
13        cin >> sueldo[i];
14    }
15
16    int sueldomax = sueldo[0];
17    for ( int i = 0; i < n; i++ ) {
18        if ( sueldo[i] > sueldomax ) {
19            sueldomax = sueldo[i];
20        }
21    }
22
23    cout << "Empleado(s) con mayor sueldo:"<< endl;
24    for ( int i = 0; i < n; i++ ) {
25        if ( sueldo[i] == sueldomax ) {
26            cout << "Empleado: "<< emp[i] << " / Sueldo: $"<< sueldo[i] << endl;
27        }
28    }
29
30    return 0;
31 }
```

```
C:\Program Files\Ampss\www  X  +  v
Sueldo que tiene: 2400
Nombre del empleado 10: Elena
Sueldo que tiene: 1700
Nombre del empleado 11: Daniel
Sueldo que tiene: 1800
Nombre del empleado 12: Andrea
Sueldo que tiene: 2500
Nombre del empleado 13: Miguel
Sueldo que tiene: 2400
Nombre del empleado 14: Marta
Sueldo que tiene: 1700
Nombre del empleado 15: Pablo
Sueldo que tiene: 2500
Nombre del empleado 16: Isabel
Sueldo que tiene: 1900
Nombre del empleado 17: David
Sueldo que tiene: 2400
Nombre del empleado 18: Natalia
Sueldo que tiene: 2000
Nombre del empleado 19: Sergio
Sueldo que tiene: 2500
Empleado(s) con mayor sueldo:
Empleado: Pedro / Sueldo: $2500
Empleado: Andrea / Sueldo: $2500
Empleado: Pablo / Sueldo: $2500
Empleado: Sergio / Sueldo: $2500
-----
```

6. Dado 5 números y un divisor, determinar cuantos números múltiplos hay el divisor en los 5 números ingresados.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int nums[] = {4, 6, 10, 15, 8};
6     int div = 3;
7
8     for ( int i = 0; i < 5; i++ ) {
9         if ( nums[i] %div == 0 ) {
10            cout << "Números multiplos de 3: " << nums[i] << endl;
11        }
12    }
13
14    return 0;
15 }
```

```
C:\Program Files\Ampss\www  X  +  v
N.meros multiplos de 3: 6
N.meros multiplos de 3: 15
-----
Process exited after 0.1104 seconds
Presione una tecla para continuar .
```



7. Escriba un programa que reciba como entrada un arreglo unidimensional ordenado de enteros (posiblemente repetidos) y genere como salida una lista de los números enteros, pero sin repeticiones.

```
9  for ( int i = 0; i < n; i++ ) {
10     cout << "Posición "<< i <<": ";
11     cin >> ar[i];
12 }
13
14 for ( int i = 1; i < n; i++ ) { // el primer for i y el segundo j
15     for ( int j = 0; j < n-1; j++ ) {
16         if ( ar[j] > ar[i] ) {
17             int temp = ar[i];
18             ar[i] = ar[j];
19             ar[j] = temp;
20         }
21     }
22 }
23 cout << "Arreglo ordenado: " << endl;
24 for ( int i : ar ) {
25     cout << i << " ";
26 }
27
28 int noreps[n];
29 int contarnoreps = 0;
30 for ( int i = 0; i < n; i++ ) {
31     bool existe = false;
32     for ( int j = 0; j < contarnoreps; j++ ) {
33         if ( ar[i] == noreps[j] ) {
34             existe = true;
35             break;
36         }
37     }
38     if ( !existe ) {
39         noreps[contarnoreps++] = ar[i];
40     }
41 }
42
43 cout << endl;
44 cout << "Arreglo sin repeticiones:" << endl;
45 for ( int i = 0; i < contarnoreps; i++ ) {
46     cout << noreps[i] << " ";
47 }
48
49 return 0;
```

```
C:\Program Files\Ampps\www X + v
Ingrese los valores numéricos para el arreglo:
Posición 0: 2
Posición 1: 5
Posición 2: 6
Posición 3: 2
Posición 4: 4
Posición 5: 6
Posición 6: 8
Posición 7: 7
Posición 8: 6
Posición 9: 3
Arreglo ordenado:
2 2 3 4 5 6 6 7 8
Arreglo sin repeticiones:
2 3 4 5 6 7 8

Process exited after 12.03 seconds with return value 0
Presione una tecla para continuar . . .
```

8. Lea 4 números y almacénelos en un arreglo llamado "A", y otros 4 números en un arreglo llamado "B", determine cuantos números de A se encuentran en "B".

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n = 4;
6      int A[] = {2, 5, 3, 7};
7      int B[] = {7, 4, 9, 3};
8
9      int contar = 0;
10     for ( int i = 0; i < n; i++ ) {
11         for ( int j = 0; j < n; j++ ) {
12             if ( A[i] == B[j] ) {
13                 contar++;
14             }
15         }
16     }
17
18     cout << "Hay '" << contar << "' números de A en B";
19
20     return 0;
21 }
22
```

```
C:\Program Files\Ampps\www X + v
Hay '2' números de A en B

Process exited after 0.07569 seconds with return value 0
Presione una tecla para continuar . . .
```

9. Crear un programa que lea la cantidad de números deseada por el cliente y no permita introducir números repetidos.



```
3
4 int main() {
5     int n;
6     cout << "Elija el tamaño del arreglo: ";
7     cin >> n;
8     int ar[n];
9
10    for ( int i = 0; i < n; i++ ) {
11        cout << "Ingrese el número " << i << " del arreglo: ";
12        int num;
13        cin >> num;
14
15        bool existe = false;
16        for ( int ii = 0; ii < i; ii++ ) {
17            if ( ar[ii] == num ) {
18                existe = true;
19                break;
20            }
21        }
22
23        if ( existe ) {
24            cout << "Este número ya está en el arreglo, introduzca otro número: ";
25            i--;
26        } else {
27            ar[i] = num;
28        }
29    }
30
31    cout << "Arreglo: ";
32    for ( int i : ar ) {
33        cout << i << " ";
34    }
35
36    return 0;
37 }
```

```
C:\Program Files\Ampps\www X + v
Elija el tamaño del arreglo: 5
Ingrese el número 0 del arreglo: 1
Ingrese el número 1 del arreglo: 2
Ingrese el número 2 del arreglo: 2
Este número ya está en el arreglo, introduzca otro número:
Ingrese el número 2 del arreglo: 3
Ingrese el número 3 del arreglo: 4
Ingrese el número 4 del arreglo: 5
Arreglo: 1 2 3 4 5
-----
Process exited after 8.858 seconds with return value 0
Presione una tecla para continuar . . . |
```

10. Desarrollar un programa que rellene un arreglo con los 100 primeros números enteros y los muestre en pantalla en orden ascendente. Menor a mayor.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int ar[100];
6
7     for ( int i = 0; i < 100; i++ ) {
8         ar[i] = i+1;
9     }
10
11    cout << "Números enteros del 1 al 100" << endl;
12    for ( int i : ar ) {
13        cout << i << endl;
14    }
15
16    return 0;
17 }
18
```

```
C:\Program Files\Ampps\www X + v
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
-----
```

11. Crear un programa que lea 100 números por teclado y los ordene descendientemente. Mayor a menor



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 100;
6     int ar[n];
7
8     cout << "Ingrese los números del 1 al 100:" << endl;
9     for ( int i = 0; i < n; i++ ) {
10         cout << "Posición "<< i << ": ";
11         cin >> ar[i];
12     }
13
14     for (int i = 0; i < n-1; i++) {
15         for (int j = 0; j < n-1-i; j++) {
16             if (ar[j] < ar[j+1]) {
17                 int temp = ar[j];
18                 ar[j] = ar[j+1];
19                 ar[j+1] = temp;
20             }
21         }
22     }
23
24     cout << "Arreglo en orden descendente:" << "\n\n";
25     for ( int i : ar ) {
26         cout << i << endl;
27     }
28
29     return 0;
30 }
```

26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

Process exited after 0.0964
Presione una tecla para con

12. Programa que llena un arreglo con los números impares en el rango del 1 al 100.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 100;
6     int ar[n];
7
8     for ( int i = 0; i < n; i++ ) {
9         ar[i] = i;
10     }
11
12     cout << "Números impares del rango de 1 al 100:" << "\n\n";
13     for ( int i : ar ) {
14         if ( ar[i] % 2 != 0 ) {
15             cout << i << endl;
16         }
17     }
18
19     return 0;
20 }
21
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 50;
6     int ar[n];
7
8     for ( int i = 0, j = 1; i < n; i++, j += 2 ) {
9         ar[i] = j;
10     }
11
12     cout << "Números impares del rango de 1 al 100:" << "\n\n";
13     for ( int i : ar ) {
14         cout << i << endl;
15     }
16
17     return 0;
18 }
19
```

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n = 100;
7     vector<int> ar;
8
9     for ( int i = 1; i < 100; i += 2 ) {
10         ar.push_back(i);
11     }
12
13     cout << "Números impares del rango de 1 al 100:" << "\n\n";
14     for ( int i : ar ) {
15         cout << i << endl;
16     }
17
18     return 0;
19 }
20
```

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n = 100;
7     vector<int> ar;
8
9     for ( int i = 1; i < 100; i++ ) {
10         if ( i % 2 != 0 ) {
11             ar.push_back(i);
12         }
13     }
14
15     cout << "Números impares del rango de 1 al 100:" << "\n\n";
16     for ( int i : ar ) {
17         cout << i << endl;
18     }
19
20     return 0;
21 }
22
```



```
C:\Program Files\Ampps\www X + v C:\Program Files\Ampps\www X + v
Números impares del rango de 1 al 100:
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
49
51
53
55
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99
-----
Process exited after 0.09497 seconds
Presione una tecla para continuar
```

13. desarrollar un programa que lea 10 números por teclado los almacenen en un arreglo y muestre la suma resta multiplicación y división de todos.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 10;
6     int nums[n];
7     int suma = 0, mult = 1;
8     double div = 0;
9
10    for ( int i = 0; i < n; i++ ) {
11        cout << "Número " << i << ": ";
12        cin >> nums[i];
13        suma = suma+nums[i];
14        mult = mult*nums[i];
15        div = static_cast<double>(suma)/(i+1);
16    }
17
18    cout << "Suma: " << suma << endl;
19    cout << "Multiplicación: " << mult << endl;
20    cout << "División: " << div << endl;
21
22    return 0;
23 }
24
C:\Program Files\Ampps\www X + v
Número 0: 1
Número 1: 2
Número 2: 5
Número 3: 6
Número 4: 3
Número 5: 8
Número 6: 9
Número 7: 4
Número 8: 6
Número 9: 5
Suma: 49
Multiplicación: 1555200
División: 4.9
-----
Process exited after 6.037 seconds
Presione una tecla para continuar
```



14. Realizar un programa que encuentre los números impares del 1 al 100 y los ordene y muestre en pantalla descendente.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 100;
6     int ar[n];
7
8     for ( int i = 0; i < n; i++ ) {
9         ar[i] = i+1;
10    }
11
12    for (int i = 0; i < n-1; i++) {
13        for (int j = 0; j < n-1-i; j++) {
14            if (ar[j] < ar[j+1]) {
15                swap( ar[j], ar[j+1] );
16            }
17        }
18    }
19
20    cout << "Números impares del rango de 1 al 100 descendente:" << "\n\n";
21    for ( int i : ar ) {
22        if ( ar[i] % 2 != 0 ) {
23            cout << i << endl;
24        }
25    }
26
27    return 0;
28 }
```

Números impares del rango de 1 al 100 descendente:

99
97
95
93
91
89
87
85
83
81
79
77
75
73
71
69
67
65
63
61
59
57
55
53
51
49
47
45

15. Realizar un programa que genere 100 numeros al azar y los ordene descendente

```
1 #include <iostream>
2 #include <ctime>
3 using namespace std;
4
5 int main() {
6     int n = 100;
7     int nums[n];
8
9     srand(time(0));
10    for ( int i=0; i<n; i++ ) {
11        nums[i] = rand() % 101;
12    }
13
14    for (int i = 0; i < n-1; i++) {
15        for (int j = 0; j < n-1-i; j++) {
16            if (nums[j] < nums[j+1]) {
17                swap( nums[j], nums[j+1] );
18            }
19        }
20    }
21
22    cout << "Números aleatorios ordenados descendente:" << "\n\n";
23    for ( int i : nums ) {
24        cout << i << endl;
25    }
26
27    return 0;
28 }
29
```

Números ordenados descendente:

99
99
99
99
97 100
96 99
95 98
95 97
94 97
94 96
94 94
92 93
90 92
87 92
86 91
85 91
83 90
83 89
83 88
80 88
80 87
78 85
77 81
76 79
75 79
74 76
74 74
72 72
72 72
71 71
70 70
69 69

Resources Compile Log Debug Find Results Close

compilation
- Errors: 0
- Warnings: 0

16. Realice un programa que tome 10 numeros y los ordene acendente



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 10;
6     int ar[n];
7
8     cout << "Ingrese los números:" << endl;
9     for ( int i = 0; i < n; i++ ) {
10         cout << "Posición "<< i << ": ";
11         cin >> ar[i];
12     }
13
14     for (int i = 0; i < n-1; i++) {
15         for (int j = 0; j < n-1-i; j++) {
16             if (ar[j] > ar[j+1]) {
17                 int temp = ar[j];
18                 ar[j] = ar[j+1];
19                 ar[j+1] = temp;
20             }
21         }
22     }
23
24     cout << "\nArreglo en orden ascendente:" << "\n\n";
25     for ( int i : ar ) {
26         cout << i << endl;
27     }
28
29     return 0;
30 }
```

C:\Program Files\Ampmps\www × +

Ingrese los números:

Posición 0: 5
Posición 1: 1
Posición 2: 6
Posición 3: 8
Posición 4: 9
Posición 5: 3
Posición 6: 4
Posición 7: 2
Posición 8: 7
Posición 9: 5

Arreglo en orden ascendente:

1
2
3
4
5
5
6
7
8
9

Process exited after 7.036 seco
Presione una tecla para continu

17. Desarrollar un programa que lea 10 números por teclado 5 para un arreglo y 5 para otro arreglo distinto mostrar los 10 números en pantalla mediante un solo arreglo



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 5;
6     int ar[n], ar2[n];
7
8     cout << "Ingrese los números Arreglo 1:" << endl;
9     for ( int i = 0; i < n; i++ ) {
10         cout << "Posición "<< i << ": ";
11         cin >> ar[i];
12     }
13     cout << "Ingrese los números Arreglo 2:" << endl;
14     for ( int i = 0; i < n; i++ ) {
15         cout << "Posición "<< i << ": ";
16         cin >> ar2[i];
17     }
18
19     int nn = sizeof(ar) / sizeof(ar[0]) + sizeof(ar2) / sizeof(ar2[0]);
20     int ar3[nn];
21
22     for (int i = 0; i < n; i++) {
23         ar3[i] = ar[i];
24     }
25     for (int i = 0; i < n; i++) {
26         ar3[i+n] = ar2[i];
27     }
28
29     cout << "\nArreglo en orden ascendente:" << "\n\n";
30     for ( int i : ar3 ) {
31         cout << i << " ";
32     }
33
34     return 0;
35 }
```

C:\Program Files\Ampss\www X + v

Ingrese los números Arreglo 1:
Posición 0: 4
Posición 1: 5
Posición 2: 6
Posición 3: 1
Posición 4: 2
Ingrese los números Arreglo 2:
Posición 0: 3
Posición 1: 9
Posición 2: 8
Posición 3: 7
Posición 4: 6

Arreglo en orden ascendente:

4
5
6
1
2
3
9
8
7
6

Process exited after 6.685 seconds
Presione una tecla para continuar

18. Desarrolle un programa que lea 5 números por teclado los copie a otro arreglo multiplicado por 2 y muestre todos los datos ordenados ascendente usando un 3er arreglo.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 5;
6     int ar[n], ar2[n], ar3[n];
7
8     cout << "Ingrese los números Arreglo 1:" << endl;
9     for ( int i = 0; i < n; i++ ) {
10         cout << "Posición "<< i << ": ";
11         cin >> ar[i];
12         ar2[i] = ar[i]*2;
13         ar3[i] = ar2[i];
14     }
15
16     for (int i = 0; i < n-1; i++) {
17         for (int j = 0; j < n-1-i; j++) {
18             if (ar3[j] < ar3[j+1]) {
19                 swap( ar3[j], ar3[j+1] );
20             }
21         }
22     }
23
24     cout << "\nArreglo en orden descendente:" << "\n\n";
25     for ( int i : ar3 ) {
26         cout << i << " ";
27     }
28
29     return 0;
30 }
```

C:\Program Files\Ampss\www X + v

Ingrese los números Arreglo 1:
Posición 0: 1
Posición 1: 5
Posición 2: 9
Posición 3: 3
Posición 4: 2

Arreglo en orden descendente:

18 10 6 4 2

Process exited after 8.865 seconds
Presione una tecla para continuar



19. Desarrollar un programa que mediante un arreglo almacene números tanto positivos como negativos y los muestre ordenados ascendentemente.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n = 6;
6     int ar[n];
7
8     cout << "Ingrese los números Arreglo:" << endl;
9     for ( int i = 0; i < n; i++ ) {
10         cout << "Posición "<< i <<": ";
11         cin >> ar[i];
12     }
13
14     for (int i = 0; i < n-1; i++) {
15         for (int j = 0; j < n-1-i; j++) {
16             if (ar[j] > ar[j+1]) {
17                 swap( ar[j], ar[j+1] );
18             }
19         }
20     }
21
22     cout << endl;
23     cout << "Arreglo en orden ascendente:" << "\n\n";
24     for ( int i : ar ) {
25         cout << i << " ";
26     }
27
28     return 0;
29 }
```

C:\Program Files\Ampps\www X +

Ingrese los números Arreglo:
Posición 0: 6
Posición 1: -2
Posición 2: 4
Posición 3: 3
Posición 4: -9
Posición 5: 1

Arreglo en orden ascendente:
-9 -2 1 3 4 6

Process exited after 14.76 seconds
Presione una tecla para continuar

20. Desarrolle un programa que rellene un arreglo con 20 números y luego busque un número en concreto y devuelva la posición del mismo.

```
1 #include <iostream>
2 #include <ctime>
3 using namespace std;
4
5 int main() {
6     int n = 20;
7     int nums[n];
8
9     srand(time(0));
10    for ( int i=0; i<n; i++ ) {
11        nums[i] = rand() % 21;
12    }
13
14    int pos = 0;
15    for ( int i=0; i<20; i++ ) {
16        if ( nums[i]==10 ) {
17            pos = i;
18            break;
19        }
20    }
21
22    cout << "Arreglo: " << endl;
23    for ( int i : nums ) {
24        cout << i << " ";
25    }
26    cout << endl;
27    cout << "Número 10 encontrado en la posición: " << pos;
28
29    return 0;
30 }
31
```

C:\Program Files\Ampps\www X +

Arreglo:
1 9 8 3 4 10 17 11 0 1 17 6 19 10 3 6 6 15 4 16
Número 10 encontrado en la posición: 5

Process exited after 0.08048 seconds with return val
Presione una tecla para continuar . . . |

21. Construya un programa que realice las siguientes acciones. Considere un arreglo ordenado en forma ascendente
- Genere un arreglo de 100 números aleatorios entre 0 y 100



- Dado un número ingresado por el usuario, de como resultado la posición de la primera ocurrencia.
- Dado un número ingresado ver cuantas veces se repite en el arreglo.
- Muestre el arreglo las veces que el usuario lo requiera.

```
7 int nums[n];
8
9 srand(time(0));
10 for (int i=0; i<n; i++) {
11     nums[i] = rand() % 101;
12 }
13
14 int buscar;
15 cout << "¿Qué número quieres buscar?" << endl;
16 cin >> buscar;
17 int mostrar;
18 cout << "¿Cuántas veces quieres mostrar el arreglo?" << endl;
19 cin >> mostrar;
20
21 int pos = 0;
22 int contar = 0;
23 for (int i=0; i<20; i++) {
24     if (nums[i]==buscar) {
25         pos = i;
26         contar++;
27     }
28 }
29
30 cout << endl;
31 cout << "Número encontrado en la posición: " << pos << endl;
32 cout << contar << " veces" << endl;
33
34 for (int j = 0; j < mostrar; j++) {
35     cout << "Arreglo: " << endl;
36     for (int i : nums) {
37         cout << i << " ";
38     }
39     cout << endl;
40 }
```

```
C:\Program Files\Ampps\www x + v
¿Qué n-mero quieres buscar?
20
¿Cuántas veces quieres mostrar el arreglo?
2

Número encontrado en la posición: 9
1 veces
Arreglo:
8 99 92 90 59 28 30 62 19 20 86 83 12 33 50 92 94 58 80 80 45 86 82 4 95 3 9 42 32 33 88 39 58 100 9 91 42 23 49 1 96 95
81 86 23 43 99 14 85 34 31 76 95 96 94 5 33 12 28 35 49 21 6 18 30 62 48 93 21 40 35 2 98 33 67 32 62 58 84 10 60 12 51
71 90 60 81 61 15 96 12 32 12 56 21 32 68 15 41 70
Arreglo:
8 99 92 90 59 28 30 62 19 20 86 83 12 33 50 92 94 58 80 80 45 86 82 4 95 3 9 42 32 33 88 39 58 100 9 91 42 23 49 1 96 95
81 86 23 43 99 14 85 34 31 76 95 96 94 5 33 12 28 35 49 21 6 18 30 62 48 93 21 40 35 2 98 33 67 32 62 58 84 10 60 12 51
71 90 60 81 61 15 96 12 32 12 56 21 32 68 15 41 70

Process exited after 1.246 seconds with return value 0
Presione una tecla para continuar . . . |
```

22. Una compañía distribuye N productos a distintos comercios de la ciudad. Para ello almacena en un arreglo toda la información relacionada con su mercancía:

- Clave
- Descripción
- Existencia
- Mínimo para mantener de existencia.
- Precio Unitario.



```
#include <iostream>

using namespace std;

int main() {
    int cant;
    cout << "Ingrese la cantidad de productos: ";
    cin >> cant;

    string clave[cant];
    string descripcion[cant];
    int existencia[cant];
    int minimo[cant];
    float precioUnitario[cant];

    for (int i = 0; i < cant; i++) {
        cout << "Ingrese la clave del producto " << i << ": ";
        cin >> clave[i];

        cout << "Ingrese la descripcion del producto " << i << ": ";
        cin >> descripcion[i];

        cout << "Ingrese la existencia del producto " << i << ": ";
        cin >> existencia[i];

        cout << "Ingrese el minimo para mantener de existencia del producto " << i << ": ";
        cin >> minimo[i];

        cout << "Ingrese el precio unitario del producto " << i << ": ";
        cin >> precioUnitario[i];
    }

    for (int j = 0; j < cant; j++) {
        cout << "Datos del producto " << j
            << "\nClave: " << clave[j] << " "
            << "Descripcion: " << descripcion[j] << " "
            << "Existencia: " << existencia[j] << " "
            << "Minimo para mantener de existencia: " << minimo[j] << " "
            << "Precio Unitario: $" << precioUnitario[j] << "\n";
    }
    return 0;
}
```

```
Ingrese la cantidad de productos: 3
Ingrese la clave del producto 0: 111
Ingrese la descripcion del producto 0: coca
Ingrese la existencia del producto 0: 30
Ingrese el minimo para mantener de existencia del producto 0: 1
Ingrese el precio unitario del producto 0: 20
Ingrese la clave del producto 1: 222
Ingrese la descripcion del producto 1: gansito
Ingrese la existencia del producto 1: 50
Ingrese el minimo para mantener de existencia del producto 1: 1
Ingrese el precio unitario del producto 1: 16
Ingrese la clave del producto 2: 333
Ingrese la descripcion del producto 2: galletas
Ingrese la existencia del producto 2: 26
Ingrese el minimo para mantener de existencia del producto 2: 1
Ingrese el precio unitario del producto 2: 22
Datos del producto 0
Clave: 111 Descripcion: coca Existencia: 30 Minimo para mantener de existencia: 1 Precio Unitario: $20
Datos del producto 1
Clave: 222 Descripcion: gansito Existencia: 50 Minimo para mantener de existencia: 1 Precio Unitario: $16
Datos del producto 2
Clave: 333 Descripcion: galletas Existencia: 26 Minimo para mantener de existencia: 1 Precio Unitario: $22
```

23. Escriba un programa que efectúe las siguientes operaciones:

- Venta de un producto: se deben actualizar los campos que correspondan y verificar que la nueva existencia no esté por debajo del mínimo. (Datos: clave, cantidad vendida)



UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

- b) Reabastecimiento de un producto: se deben actualizar los campos que correspondan. (Datos: clave, cantidad comprada.)
- c) Actualizar el precio de un producto. (Datos: clave, porcentaje de aumento.)
- d) Informar sobre un productor: se deben proporcionar todos los datos relacionados con un producto. (Dato: clave)



```
#include <iostream>
#include <vector>

using namespace std;

int main() {

    int cantidad;
    cout << "Ingrese la cantidad de productos: ";
    cin >> cantidad;

    string producto[cantidad];
    vector <string> clave(cantidad);
    int cant_prod[cantidad];
    float precio[cantidad];

    for (int i = 0; i < cantidad; i++){
        cout << "Ingrese el nombre del producto " << i << ": ";
        cin>>producto[i];

        cout << "Ingrese la clave del producto "<< producto[i] << ": ";
        cin>>clave[i];

        cout << "Ingrese la cantidad del producto " << producto[i] << ": ";
        cin>>cant_prod[i];

        cout << "Ingrese el precio del producto " << producto[i] << ": ";
        cin >> precio[i];
    }

    string clave_venta;
    int cantidad_venta;

    cout << "Ingrese la clave del producto que desea vender: ";
    cin >> clave_venta;
    cout << "Ingrese la cantidad vendida del producto: ";
    cin >> cantidad_venta;

    for (int i = 0; i < clave.size(); i++) {
        if (clave[i] == clave_venta) {
            if (cant_prod[i] >= cantidad_venta) {
                cant_prod[i] -= cantidad_venta;
            }
        }
    }
}
```



```
        cout << "Producto vendido: " << producto[i] << "\n";
        cout << "Cantidad restante: " << cant_prod[i] << "\n";
        if (cant_prod[i] < 5) {
            cout << "El stock esta por debajo del minimo.\n";
        }
    } else {
        cout << "No hay suficiente stock para la venta.\n";
    }
}

string clave_reabastecer;
int cantidad_reabastecer;
cout << "Ingrese la clave del producto que desea reabastecer: ";
cin >> clave_reabastecer;
cout << "Ingrese la cantidad comprada del producto: ";
cin >> cantidad_reabastecer;

for (int i = 0; i < clave.size(); i++) {
    if (clave[i] == clave_reabastecer) {
        cant_prod[i] += cantidad_reabastecer;
        cout << "Producto reabastecido: " << producto[i] << "\n";
        cout << "Cantidad actual: " << cant_prod[i] << "\n";
    }
}

string clave_precio;
float porcentaje_aumento;
cout << "Ingrese la clave del producto para actualizar el precio: ";
cin >> clave_precio;
cout << "Ingrese el porcentaje de aumento de precio: ";
cin >> porcentaje_aumento;

for (int i = 0; i < clave.size(); i++) {
    if (clave[i] == clave_precio) {
        precio[i] += precio[i] * (porcentaje_aumento / 100);
        cout << "Nuevo precio del producto " << producto[i] << ": $" << precio[i] << "\n";
    }
}

string clave_info;
cout << "Ingrese la clave del producto para obtener informacion: ";
cin >> clave_info;

for (int i = 0; i < clave.size(); i++) {
    if (clave[i] == clave_info) {
        cout << "\nInformacion del producto:\n";
        cout << "Nombre: " << producto[i] << "\n";
        cout << "Clave: " << clave[i] << "\n";
        cout << "Cantidad en stock: " << cant_prod[i] << "\n";
        cout << "Precio: $" << precio[i] << "\n";
    }
}

return 0;
}
```




```
Ingrese la cantidad de productos: 1
Ingrese el nombre del producto 0: coca
Ingrese la clave del producto coca: 111
Ingrese la cantidad del producto coca: 30
Ingrese el precio del producto coca: 20
Ingrese la clave del producto que desea vender: 111
Ingrese la cantidad vendida del producto: 7
Producto vendido: coca
Cantidad restante: 23
Ingrese la clave del producto que desea reabastecer: 111
Ingrese la cantidad comprada del producto: 2
Producto reabastecido: coca
Cantidad actual: 25
Ingrese la clave del producto para actualizar el precio: 111
Ingrese el porcentaje de aumento de precio: 20
Nuevo precio del producto coca: $24
Ingrese la clave del producto para obtener informacion: 111

Informacion del producto:
Nombre: coca
Clave: 111
Cantidad en stock: 25
Precio: $24
```

24. Al momento de su ingreso al hospital, a un paciente se le solicitan los siguientes datos:

- Nombre
- Edad
- Sexo
- Domicilio:
 - Calle
 - Número
 - Ciudad
- Teléfono
- Seguro (este campo tendrá el valor VERDADERO si el paciente tiene seguro médico y FALSO en otro caso).

Escriba un programa que pueda llevar a cabo las siguientes operaciones:

- a) Listar los nombres de todos los pacientes hospitalizados.
- b) Obtener el porcentaje de pacientes hospitalizados en las siguientes categorías (dadas por la edad):
 - a. Niños: hasta 13 años.
 - b. Jóvenes: mayores de 13 años y menores de 30
 - c. Adultos: mayores de 30 años.
- c) Obtener el porcentaje de hombres y de mujeres hospitalizados.
- d) Dado el nombre de un paciente, listar todos los datos relacionados con dicho paciente.



```
#include <iostream>
using namespace std;
int main() {
    int cantidad;
    cout << "Ingrese la cantidad de pacientes: ";
    cin >> cantidad;

    string nombres[cantidad];
    int edad[cantidad];
    string sexo[cantidad];
    string domicilio[cantidad];
    string calle[cantidad];
    int numero[cantidad];
    string ciudad[cantidad];
    string telefono[cantidad];
    string seguro[cantidad];

    for (int i = 0; i < cantidad; i++) {
        cout << "Ingrese el nombre del paciente " << i << ": ";
        cin >> nombres[i];

        cout << "Ingrese la edad del paciente " << i << ": ";
        cin >> edad[i];

        cout << "Ingrese el sexo del paciente " << i << ": (M/F) ";
        cin >> sexo[i];

        cout << "Ingrese el domicilio del paciente " << i << ": ";
        cin >> domicilio[i];

        cout << "Ingrese la calle del paciente " << i << ": ";
        cin >> calle[i];

        cout << "Ingrese el numero de casa del paciente " << i << ": ";
        cin >> numero[i];

        cout << "Ingrese la ciudad donde se encuentra el paciente " << i << ": ";
        cin >> ciudad[i];

        cout << "Ingrese el telefono del paciente " << i << ": ";
        cin >> telefono[i];

        string asegurado;
        cout << "¿El paciente " << nombres[i] << " cuenta con seguro? (s/n): ";
        cin >> asegurado;

        if (asegurado == "s" || asegurado == "S") {
            seguro[i] = "Con seguro";
        } else if (asegurado == "n" || asegurado == "N") {
            seguro[i] = "Sin seguro";
        }
    }

    int ninos = 0, jovenes = 0, adultos = 0;
    float porcen_ninos = 0, porcen_jovenes = 0, porcen_adultos = 0;
```

```

for (int i = 0; i < cantidad; i++) {
    if (edad[i] <= 13) {
        ninos++;
    } else if (edad[i] > 13 && edad[i] < 30) {
        jovenes++;
    } else {
        adultos++;
    }
}
porcen_ninos = (ninos / cantidad) * 100;
porcen_jovenes = (jovenes / cantidad) * 100;
porcen_adultos = (adultos / cantidad) * 100;

int mujeres = 0, hombres = 0;
float porcen_m = 0, porcen_h = 0;

for (int i = 0; i < cantidad; i++) {
    if (sexo[i] == "F" || sexo[i] == "f") {
        mujeres++;
    } else if (sexo[i] == "M" || sexo[i] == "m") {
        hombres++;
    }
}
porcen_m = (mujeres / cantidad) * 100;
porcen_h = (hombres / cantidad) * 100;

cout << "Nombres de los pacientes: ";
for (int i = 0; i < cantidad; i++) {
    cout << nombres[i] << " ";
}
cout << "\n--Estatus del seguro: ";
for (int i = 0; i < cantidad; i++) {
    cout << seguro[i] << " ";
}
cout << "\n--Porcentaje de Ninos: %" << porcen_ninos;
cout << "\n--Porcentaje de Jovenes: %" << porcen_jovenes;
cout << "\n--Porcentaje de Adultos: %" << porcen_adultos;
cout << "\n--Porcentaje de Mujeres: %" << porcen_m;
cout << "\n--Porcentaje de Hombres: %" << porcen_h;

string buscar;
cout << "\nIngrese el nombre del paciente a buscar: ";
cin>>buscar;
for (int i = 0; i < cantidad; i++) {
    if (nombres[i] == buscar) {
        cout << "\nEl paciente " << nombres[i]
            << " tiene " << edad[i] << " years, de sexo " << sexo[i]
            << ", con domicilio en " << domicilio[i]
            << ", calle " << calle[i] << " y numero " << numero[i]
            << ", residente de la ciudad de " << ciudad[i]
            << "\n--Telefono: " << telefono[i]
            << "\n--Seguro: " << seguro[i] << "\n";
    }
}
return 0;

```

```

Ingrese la cantidad de pacientes: 1
Ingrese el nombre del paciente 0: jareth
Ingrese la edad del paciente 0: 18
Ingrese el sexo del paciente 0: (M/F) m
Ingrese el domicilio del paciente 0: centro
Ingrese la calle del paciente 0: hidalgo
Ingrese el numero de casa del paciente 0: 55
Ingrese la ciudad donde se encuentra el paciente 0: escuiyork
Ingrese el telefono del paciente 0: 098765
¿El paciente jareth cuenta con seguro? (s/n): s
Nombres de los pacientes: jareth
--Estatus del seguro: Con seguro
--Porcentaje de Ninos: %0
--Porcentaje de Jovenes: %100
--Porcentaje de Adultos: %0
--Porcentaje de Mujeres: %0
--Porcentaje de Hombres: %100
Ingrese el nombre del paciente a buscar: jareth

El paciente jareth tiene 18 years, de sexo m, con domicilio en centro, calle hidalgo y numero 55, residente de la ciudad de escuiyork.
--Telefono: 098765.
--Seguro: Con seguro.

```

25. Supongamos que en una elección hubo 12 candidatos (con identificadores 1,2,3, ..., 12). Por otra parte, los votos para cada candidato se teclean de manera desorganizada como se muestra a continuación:



UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

1 5 7 5 1 12 10 7 1 7 5 8 1 5 -1(número negativo para detener la captura de votos)

Construya un programa que pueda proporcionar la siguiente información:

- a. El número de votos de cada candidato al final de la elección.
- b. El candidato ganador, el número de votos que obtuvo y el porcentaje correspondiente del total de la elección. Suponemos que el candidato ganador no empató en número de votos con otro candidato.



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

int main(){
    int i=0;
    vector<int> votos;
    int num;

    while(true){
        cout<<"Ingrese el numero del candidato seleccionado: (1-12) ";
        cin>>num;

        if(num>12||num==0){
            cout<<"candidato inexistente";
        }else if(num<0){
            break;
        }else{
            votos.push_back(num);
        }
    }
    vector<int> conteoVotos(12, 0);

    for (int i = 0; i < votos.size(); i++) {
        conteoVotos[votos[i] - 1]++;
    }
    int totalVotos=votos.size();

    cout << "Resultados:\n";
    for (int i = 0; i < conteoVotos.size(); i++) {
        cout << "Candidato " <<(i+1) << ": " << conteoVotos[i] << " votos\n";
    }

    int ganador;
    int max;
    for (int i = 0; i < conteoVotos.size(); i++) {
        if (conteoVotos[i] > max) {
            max = conteoVotos[i];
            ganador = i + 1;
        }
    }
    float porcentajeGanador = (static_cast<float>(max) / totalVotos) * 100;
    cout<<"Candidato ganador:"<<endl;
    cout<<"El candidato "<< ganador <<" gano con "<< max<<" votos ("<<porcentajeGanador<<"% del total).";

    return 0;
}
```



```
Ingrese el numero del candidato seleccionado: (1-12) 1
Ingrese el numero del candidato seleccionado: (1-12) 2
Ingrese el numero del candidato seleccionado: (1-12) 3
Ingrese el numero del candidato seleccionado: (1-12) 4
Ingrese el numero del candidato seleccionado: (1-12) 2
Ingrese el numero del candidato seleccionado: (1-12) 1
Ingrese el numero del candidato seleccionado: (1-12) 7
Ingrese el numero del candidato seleccionado: (1-12) 1
Ingrese el numero del candidato seleccionado: (1-12) -1
Resultados:
Candidato 1: 3 votos
Candidato 2: 2 votos
Candidato 3: 1 votos
Candidato 4: 1 votos
Candidato 5: 0 votos
Candidato 6: 0 votos
Candidato 7: 1 votos
Candidato 8: 0 votos
Candidato 9: 0 votos
Candidato 10: 0 votos
Candidato 11: 0 votos
Candidato 12: 0 votos
Candidato ganador:
El candidato 1 gano con 3 votos (37.5% del total).
```

26. Almacene en un arreglo las temperaturas de un día (0-23 horas). Calcule su media e imprimir la temperatura más alta y la más baja; asimismo imprimir la hora respectiva. Por ejemplo, la temperatura media fue de 21.5 grados Celsius, la más alta de 29 grados Celsius a las 14 horas y la más baja de 9.4 grados Celsius a las 3 horas.



```
#include <iostream>

using namespace std;
int main() {

    float temperaturas[24];
    float suma = 0.0;

    for (int i = 0; i < 24; i++) {
        cout << "Ingrese la temperatura para la hora " << i << ": ";
        cin >> temperaturas[i];
        suma += temperaturas[i];
    }

    float temp_max = temperaturas[0];
    float horaMax = 0;
    float temp_min = temperaturas[0];
    float horaMin = 0;

    for(int i = 0; i < 24; i++) {
        if (temperaturas[i] > temp_max) {
            temp_max = temperaturas[i];
            horaMax = i;
        }
        if (temperaturas[i] < temp_min) {
            temp_min = temperaturas[i];
            horaMin = i;
        }
    }

    float temp_media = suma / 24;

    cout<<"La temperatura media fue de "<<temp_media<<" grados Celsius."<<endl;
    cout<<"La temperatura mas alta fue de "<<temp_max<<" grados Celsius a las "<<horaMax<<" horas."<<endl;
    cout<<"La temperatura mas baja fue de "<<temp_min<<" grados Celsius a las "<<horaMin<<" horas.";

    return 0;
}
```



```
Ingrese la temperatura para la hora 0: 1
Ingrese la temperatura para la hora 1: 2
Ingrese la temperatura para la hora 2: 3
Ingrese la temperatura para la hora 3: 4
Ingrese la temperatura para la hora 4: 5
Ingrese la temperatura para la hora 5: 6
Ingrese la temperatura para la hora 6: 7
Ingrese la temperatura para la hora 7: 8
Ingrese la temperatura para la hora 8: 9
Ingrese la temperatura para la hora 9: 10
Ingrese la temperatura para la hora 10: 11
Ingrese la temperatura para la hora 11: 12
Ingrese la temperatura para la hora 12: 13
Ingrese la temperatura para la hora 13: 14
Ingrese la temperatura para la hora 14: 5
Ingrese la temperatura para la hora 15: 16
Ingrese la temperatura para la hora 16: 17
Ingrese la temperatura para la hora 17: 18
Ingrese la temperatura para la hora 18: 19
Ingrese la temperatura para la hora 19: 20
Ingrese la temperatura para la hora 20: 21
Ingrese la temperatura para la hora 21: 22
Ingrese la temperatura para la hora 22: 23
Ingrese la temperatura para la hora 23: 24
La temperatura media fue de 12.0833 grados Celsius.
La temperatura mas alta fue de 24 grados Celsius a las 23 horas.
La temperatura mas baja fue de 1 grados Celsius a las 0 horas.
```

27. Almacene en un arreglo la temperatura de cada día de una determinada semana y que realice lo siguiente:
- La temperatura promedio.
 - Un arreglo que contenga las diferencias de cada temperatura con respecto al promedio.
 - La menor temperatura y el número de día en que ocurrió.
 - La mayor temperatura y el número de día en que ocurrió.



```
int main() {
    float temperaturas[7];
    string diasSemana[] = {"Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"};
    float suma = 0.0;

    for (int i = 0; i < 7; i++) {
        cout<< "Ingrese la temperatura para el día " << diasSemana[i] << ": ";
        cin>> temperaturas[i];
        suma += temperaturas[i];
    }
    float temp_promedio = suma / 7;

    float diferencias[7];
    for (int i = 0; i < 7; i++) {
        diferencias[i] = temperaturas[i] - temp_promedio;
    }
    float temp_min = temperaturas[0], temp_max = temperaturas[0];
    int diaMin = 0, diaMax = 0;

    for (int i = 1; i < 7; i++) {
        if (temperaturas[i] < temp_min) {
            temp_min = temperaturas[i];
            diaMin = i;
        }
        if (temperaturas[i] > temp_max) {
            temp_max = temperaturas[i];
            diaMax = i;
        }
    }
    cout<<"Temperatura promedio: "<< temp_promedio<< " grados Celsius."<<endl;

    cout<<"Diferencias respecto al promedio: ";
    for (int i = 0; i < 7; i++) {
        cout << "Día " << diasSemana[i] << ": " << diferencias[i] << " grados de diferencia."<<endl;
    }

    cout << "La menor temperatura fue de " << temp_min << " grados Celsius el " << diasSemana[diaMin] <<endl;
    cout << "La mayor temperatura fue de " << temp_max << " grados Celsius el " << diasSemana[diaMax];
}
```

```
Ingrese la temperatura para el día Lunes: 10
Ingrese la temperatura para el día Martes: 20
Ingrese la temperatura para el día Miercoles: 30
Ingrese la temperatura para el día Jueves: 40
Ingrese la temperatura para el día Viernes: 50
Ingrese la temperatura para el día Sabado: 60
Ingrese la temperatura para el día Domingo: 70
Temperatura promedio: 40 grados Celsius.
Diferencias respecto al promedio: Día Lunes: -30 grados de diferencia.
Día Martes: -20 grados de diferencia.
Día Miercoles: -10 grados de diferencia.
Día Jueves: 0 grados de diferencia.
Día Viernes: 10 grados de diferencia.
Día Sabado: 20 grados de diferencia.
Día Domingo: 30 grados de diferencia.
La menor temperatura fue de 10 grados Celsius el Lunes
La mayor temperatura fue de 70 grados Celsius el Domingo
```

28. Almacene en un arreglo, 20 elementos de tipo entero e imprimir:
- La suma de los elementos que ocupan posiciones pares.
 - El mayor de los elementos que ocupan posiciones impares.
 - La posición del mayor número par.



```
int main(){
    int numeros[20];
    int suma_pares = 0;
    int mayorImpar = 0;
    int posicionMayorPar = 0;
    int mayor_par = 0;

    for (int i = 0; i < 20; i++) {
        cout << "Ingrese el numero "<<i<< ": ";
        cin >> numeros[i];

        if (i % 2 == 0) {
            suma_pares += numeros[i];
        }

        if (i % 2 != 0) {
            if (mayorImpar == 0 || numeros[i] > mayorImpar) {
                mayorImpar = numeros[i];
            }
        }

        if (numeros[i] % 2 == 0) {
            if (mayor_par == 0 || numeros[i] > mayor_par) {
                mayor_par = numeros[i];
                posicionMayorPar = i;
            }
        }
    }
    cout << "Suma de los numeros en posiciones pares: "<<suma_pares<<endl;

    if (mayorImpar != 0) {
        cout << "El mayor numero en posiciones impares es: " << mayorImpar <<endl;
    } else {
        cout << "No se ingresaron numeros en posiciones impares." <<endl;
    }

    if (posicionMayorPar != -1) {
        cout << "La posición del mayor numero par es: " << posicionMayorPar <<endl;
    } else {
        cout << "No se ingresaron numeros pares." <<endl;
    }
}
```



```
Ingrese el numero 0: 1
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
Ingrese el numero 9: 10
Ingrese el numero 10: 11
Ingrese el numero 11: 12
Ingrese el numero 12: 13
Ingrese el numero 13: 14
Ingrese el numero 14: 15
Ingrese el numero 15: 16
Ingrese el numero 16: 17
Ingrese el numero 17: 18
Ingrese el numero 18: 19
Ingrese el numero 19: 20
Suma de los numeros en posiciones pares: 100
El mayor numero en posiciones impares es: 20
La posición del mayor numero par es: 19
```

29. Almacene en un arreglo, n elementos (máximo 30) e imprimir la suma:
- de números pares.
 - de números impares.
 - total de los elementos del arreglo.



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

int main(){

    int suma_par = 0;
    int suma_impar = 0;
    int suma_total = 0;
    int tam;

    cout<<"Ingrese el tamaño del arreglo (maximo 30): ";
    cin>>tam;

    int numeros[tam];
    if (tam > 30 || tam <= 0) {
        cout<<"Tamaño del arreglo invalido." <<endl;
    } else {
        for (int i = 0; i < tam; i++) {
            int num;
            cout<<"Ingrese el numero "<<i<<" ": ";
            cin>>num;
            numeros[i]=num;
            suma_total+=num;

            if (num % 2 == 0) {
                suma_par += num;
            } else {
                suma_impar += num;
            }
        }
        cout<<"Numeros ingresados: ";
        for (int i=0; i<tam;i++) {
            cout<<numeros[i]<<" ";
        }
        cout<<endl;
        cout << "Suma de numeros pares: " << suma_par <<endl;
        cout << "Suma de numeros impares: " << suma_impar <<endl;
        cout << "Suma total de los numeros: " << suma_total;

    }

    return 0;
}
```

```
Ingrese el tamaño del arreglo (maximo 30): 5
Ingrese el numero 0: 1
Ingrese el numero 1: 2
Ingrese el numero 2: 3
Ingrese el numero 3: 4
Ingrese el numero 4: 5
Numeros ingresados: 1, 2, 3, 4, 5,
Suma de numeros pares: 6
Suma de numeros impares: 9
Suma total de los numeros: 15
=====
```

30. Almacene en una lista 15 números e imprimir cuántos son ceros, cuántos son negativos y cuántos positivos. Imprimir además la suma de los negativos y la suma de los positivos.



```
#include <iostream>

using namespace std;

int main(){

    int numeros[15];
    int ceros = 0, negativos = 0, positivos = 0;
    int sum_negativos = 0, sum_positivos = 0;

    for (int i = 0; i < 15; i++) {
        cout << "Ingrese el numero " << i << ": ";
        cin>> numeros[i];

        if (numeros[i] == 0) {
            ceros++;
        } else if (numeros[i] > 0) {
            positivos++;
            sum_positivos += numeros[i];
        } else {
            negativos++;
            sum_negativos += numeros[i];
        }
    }

    cout<<"Cantidad de ceros: "<<ceros<<endl;
    cout<<"Cantidad de numeros positivos: " << positivos<<endl;
    cout<<"Cantidad de numeros negativos: " << negativos<<endl;
    cout<<"Suma de numeros positivos: " << sum_positivos<<endl;
    cout<<"Suma de numeros negativos: " << sum_negativos;

    return 0;
}
```



```
Ingrese el numero 0: 1
Ingrese el numero 1: 2
Ingrese el numero 2: 3
Ingrese el numero 3: 4
Ingrese el numero 4: 5
Ingrese el numero 5: -1
Ingrese el numero 6: -2
Ingrese el numero 7: -3
Ingrese el numero 8: -4
Ingrese el numero 9: -5
Ingrese el numero 10: 0
Ingrese el numero 11: 0
Ingrese el numero 12: 1
Ingrese el numero 13: 2
Ingrese el numero 14: 3
Cantidad de ceros: 2
Cantidad de numeros positivos: 8
Cantidad de numeros negativos: 5
Suma de numeros positivos: 21
Suma de numeros negativos: -15
```

31. Inicialice un arreglo con 10 valores. Genere un arreglo a con los números negativos y otro arreglo b con los positivos o iguales a cero.



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

int main(){
    float numeros[10];
    vector<float> negativos;
    vector<float> positivos;

    for (int i = 0; i < 10; i++) {
        cout << "Ingrese el numero " << i << ": ";
        cin>> numeros[i];

        if (numeros[i] < 0) {
            negativos.push_back(numeros[i]);
        } else {
            positivos.push_back(numeros[i]);
        }
    }

    cout << "Arreglo original: ";
    for (int i=0; i<10;i++) {
        cout<<numeros[i]<<" ";
    }
    cout <<endl;

    cout << "Arreglo de numeros negativos: ";
    for (int i=0; i<negativos.size();i++) {
        cout<<negativos[i]<<" ";
    }
    cout <<endl;

    cout<<"Arreglo de numeros positivos o iguales a cero: ";
    for (int i=0; i<positivos.size();i++) {
        cout<<positivos[i]<<" ";
    }

    return 0;
}
```

```
Ingrese el numero 0: 1
Ingrese el numero 1: 2
Ingrese el numero 2: 3
Ingrese el numero 3: 4
Ingrese el numero 4: 5
Ingrese el numero 5: -1
Ingrese el numero 6: -2
Ingrese el numero 7: -3
Ingrese el numero 8: -4
Ingrese el numero 9: 0
Arreglo original: 1, 2, 3, 4, 5, -1, -2, -3, -4, 0,
Arreglo de numeros negativos: -1, -2, -3, -4,
Arreglo de numeros positivos o iguales a cero: 1, 2, 3, 4, 5, 0,
```

32. Almacene en un arreglo n elementos de tipo real (máximo 25) e imprima los siguientes valores:



UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

- a. Máximo.
- b. Mínimo.
- c. La media de los que estén en posiciones pares.
- d. La varianza. (<http://www.disfrutalasmaticas.com/datos/desviacion-estandar.html>)



```
int main(){
    int tam;
    cout<<"Ingrese el tamaño del arreglo: (Max 25): ";
    cin>>tam;
    int numeros[tam];

    for(int i=0;i<tam;i++){
        cout<<"Ingrese el número "<<i<<": ";
        cin>>numeros[i];
    }
    int num_max=numeros[0];
    int num_min=numeros[0];

    int suma=0;
    for(int i=0;i<tam;i++){
        suma+=numeros[i];
        if(numeros[i]<num_min){
            num_min=numeros[i];
        }
        if(numeros[i]>num_max){
            num_max=numeros[i];
        }
    }
    float media=0;
    float varianza=0;

    media=suma/tam;
    for(int i=0;i<tam;i++){
        varianza+=(numeros[i]-media)*(numeros[i]-media);
    }
    cout<<"Números ingresados: "<<endl;
    varianza=varianza/tam;
    for(int i=0;i<tam;i++){
        cout<<numeros[i]<<endl;
    }

    cout<<"Número máximo: "<<num_max<<endl;
    cout<<"Número mínimo: "<<num_min<<endl;
    cout<<"Media: "<<media<<endl;
    cout<<"Varianza: "<<varianza;

    return 0;
}
```



```
Ingrese el tamaño del arreglo: (Max 25): 7
Ingrese el número 0: 1
Ingrese el número 1: 2
Ingrese el número 2: 3
Ingrese el número 3: 4
Ingrese el número 4: 5
Ingrese el número 5: 6
Ingrese el número 6: 7
Numeros ingresados:
1
2
3
4
5
6
7
Número máximo: 7
Número mínimo: 1
Media: 4
Varianza: 4
```

33. Genere aleatoriamente un arreglo de tamaño 20 con números entre 0 y 150 y genere otras tres listas con los siguientes criterios:
- Si los números están comprendidos entre 0 y 50 irán en la lista 1.
 - Si los números están comprendidos entre 51 y 100 irán en la lista 2.
 - Si los números son mayores a 101 irán en la lista 3.
- Al final imprimir las cuatro listas.

```
int main(){
    vector<int> numeros;
    for (int i=0; i<20; i++) {
        int num= rand() % 151;
        numeros.push_back(num);
    }
    vector<int> lista1;
    vector<int> lista2;
    vector<int> lista3;

    for (int i = 0; i < numeros.size(); i++){
        if (numeros[i] >= 0 && numeros[i] <= 50) {
            lista1.push_back(numeros[i]);
        } else if (numeros[i] >= 51 && numeros[i] <= 100) {
            lista2.push_back(numeros[i]);
        } else if (numeros[i] > 100) {
            lista3.push_back(numeros[i]);
        }
    }
    cout<<"Números originales: ";
    for(int i = 0; i < 20; i++){
        cout<<numeros[i]<<" ";
    }
    cout<<endl;

    cout<<"Lista 1, de 0 a 50: ";
    for(int i = 0; i < lista1.size() ; i++){
        cout<< lista1[i]<<" ";
    }
    cout<<endl;
    cout<<"Lista 2, de 51 a 100: ";
    for(int i = 0; i < lista2.size() ; i++){
        cout<< lista2[i]<<" ";
    }
    cout<<endl;
    cout<<"Lista 3, mayores a 100: ";
    for(int i = 0; i < lista3.size() ; i++){
        cout<<lista3[i]<<" ";
    }

    return 0;
}
```

```

Numeros originales: 41 45 143 75 143 20 2 64 84 2 118 59 27 66 146 38 126 13 146 0
Lista 1, de 0 a 50: 41 45 20 2 2 27 38 13 0
Lista 2, de 51 a 100: 75 64 84 59 66
Lista 3, mayores a 100: 143 143 118 146 126 146
```

34. Almacene en un arreglo 10 números y cambie algún número por otro del mismo arreglo, realice el cambio y muestre el arreglo modificado.



```
#include <iostream>

using namespace std;

int main(){
    int numeros[10];

    for (int i = 0; i < 10; i++) {
        cout<<"Ingrese el numero "<<i<<": ";
        cin>> numeros[i];
    }

    cout << "Números Ingresados: " << endl;
    for (int i=0;i<10;i++) {
        cout<<numeros[i]<<" ";
    }
    cout<<endl;

    int cambiar_num, new_num;
    cout << "Ingrese el índice (de 0 a 9) del numero que desea cambiar: ";
    cin >> cambiar_num;
    cout << "Ingrese el índice (de 0 a 9) del numero que desea usar para el cambio: ";
    cin >> new_num;

    int indice = numeros[cambiar_num];
    numeros[cambiar_num] = numeros[new_num];
    numeros[new_num] = indice;

    cout <<"Numeros cambiados: " << endl;
    for (int i=0;i<10;i++) {
        cout<<numeros[i]<<" ";
    }

    return 0;
}
```

```
Ingrese el numero 0: 1
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
Ingrese el numero 9: 10
Números Ingresados:
1 2 3 4 5 6 7 8 9 10
Ingrese el índice (de 0 a 9) del numero que desea cambiar: 0
Ingrese el índice (de 0 a 9) del numero que desea usar para el cambio: 9
Numeros cambiados:
10 2 3 4 5 6 7 8 9 1
```

35. Genere aleatoriamente un arreglo de tamaño 20 con números comprendidos entre 1 y 20 e imprimir:
- Suma de los elementos.
 - Promedio de los elementos.
 - Varianza y desviación estándar de los valores.
 - Elemento que más se repite (moda).



```
int main() {
    vector<int> numeros(20);
    vector<int> frecuencia(21, 0);
    float suma = 0;

    for (int i = 0; i < 20; i++) {
        numeros[i] = rand() % 21;
        suma += numeros[i];
        frecuencia[numeros[i]]++;
    }
    cout << "Numeros aleatorios: ";
    for (int i = 0; i < numeros.size(); i++) {
        cout << numeros[i] << " ";
    }
    cout << endl;
    float promedio = suma / numeros.size();
    cout << "Suma de los numeros: " << suma << endl;
    cout << "Promedio de los numeros: " << promedio << endl;

    float varianza = 0;
    for (int i = 0; i < numeros.size(); i++) {
        varianza += pow(numeros[i] - promedio, 2);
    }
    varianza /= numeros.size();
    cout << "Varianza: " << varianza << endl;

    float desviacion_estandar = sqrt(varianza);
    cout << "Desviacion estandar: " << desviacion_estandar << endl;

    int moda = 0;
    int max_frecuencia = 0;
    for (int i = 0; i < 20; i++) {
        if (frecuencia[i] > max_frecuencia) {
            max_frecuencia = frecuencia[i];
            moda = i;
        }
    }
    cout << "El numero que mas se repite es " << moda << " con " << max_frecuencia << " repeticiones" << endl;
    return 0;
}
```

Numeros aleatorios: 20 8 13 19 17 16 12 0 19 20 14 5 13 6 7 8 13 14 18 18
Suma de los numeros: 260
Promedio de los numeros: 13
Varianza: 30.8
Desviacion estandar: 5.54977
El numero que mas se repite es 13 con 3 repeticiones

36. Almacene en un arreglo de tamaño 10 números reales. Calcule el promedio e indique cuántos elementos del arreglo son mayores que el promedio y genere otro arreglo con los menores o iguales.

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

int main() {
    vector <float> numeros;
    float num;

    float suma = 0;
    vector <float> menores_iguales;
    float mayores_promedio = 0;

    for(int i = 0; i < 10; i++) {
        cout<<"Ingrese el numero "<<i<<": ";
        cin>>num;
        numeros.push_back(num);
        suma += num;
    }

    float promedio = suma / numeros.size();
    for (int i = 0; i < 10; i++) {
        if (numeros[i] > promedio) {
            mayores_promedio++;
        } else {
            menores_iguales.push_back(numeros[i]);
        }
    }
    cout<<"Numeros ingresados: ";
    for(int i=0;i<10;i++){
        cout<<numeros[i]<<" ";
    }
    cout<<endl;
    cout<<"Promedio: "<<promedio<<endl;

    cout<<"Cantidad de numeros mayores al promedio: "<<mayores_promedio<<endl;

    cout<<"Numeros menores o iguales al promedio: ";
    for(int i=0;i<menores_iguales.size();i++){
        cout<<menores_iguales[i]<<" ";
    }

    return 0;
}
```

```
Ingrese el numero 0: 1
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
Ingrese el numero 9: 10
Numeros ingresados: 1 2 3 4 5 6 7 8 9 10
Promedio: 5.5
Cantidad de numeros mayores al promedio: 5
Numeros menores o iguales al promedio: 1 2 3 4 5
```

37. Almacene en un arreglo de n números enteros (máximo 30) y determine cuántos números terminan en dígito primo.



```
#include <iostream>

using namespace std;

int main() {
    int tam;
    cout<<"Ingrese la cantidad de numeros enteros a almacenar (maximo 30): ";
    cin>>tam;

    int num_primos = 0;
    int numeros[tam];

    for (int i = 0; i < tam; i++) {
        cout<<"Ingrese el numero entero "<<i<<": ";
        cin>> numeros[i];

        float ultimo_digito = numeros[i] % 10;

        if (ultimo_digito == 2 || ultimo_digito == 3 || ultimo_digito == 5 || ultimo_digito == 7) {
            num_primos++;
        }
    }
    cout<<"Numeros ingresados: ";
    for(int i=0;i<tam;i++){
        cout<<numeros[i]<<" ";
    }
    cout<<endl<<"Cantidad de numeros que terminan en digito primo: "<<num_primos;

    return 0;
}
```

Ingrese la cantidad de numeros enteros a almacenar (maximo 30): 6
Ingrese el numero entero 0: 1
Ingrese el numero entero 1: 2
Ingrese el numero entero 2: 3
Ingrese el numero entero 3: 4
Ingrese el numero entero 4: 5
Ingrese el numero entero 5: 6
Numeros ingresados: 1 2 3 4 5 6
Cantidad de numeros que terminan en digito primo: 3

38. Convierta un número entero decimal a su equivalente en representación binaria. (http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_contenidos_2c.htm)



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

int main() {
    int num_decimal;
    cout << "Ingrese un numero entero decimal: ";
    cin >> num_decimal;

    vector<int> binario;

    int temp = num_decimal;
    while (temp > 0) {
        binario.insert(binario.begin(), temp % 2);
        temp /= 2;
    }

    cout << "El numero decimal " << num_decimal << " en binario es: ";
    for (int i=0; i<binario.size();i++) {
        cout << binario[i];
    }
    cout << endl;

    return 0;
}
```

```
Ingrese un numero entero decimal: 11
El numero decimal 11 en binario es: 1011
```

39. Almacene en un arreglo n números enteros y determine ¿cuántas veces se repite cada uno de ellos

Si $n=6$ y los elementos del arreglo son: 3,4,6,6,4,6, se imprimirá:

3=1

4=2

6=3



```
#include <iostream>
#include <vector>
#include <map>

using namespace std;

int main() {

    int tam;
    cout << "Ingrese el tamaño del arreglo: ";
    cin >> tam;
    int numeros[tam];
    map<int, int> repeticiones;

    for (int i = 0; i < tam; i++) {
        cout << "Ingrese el número " << i + 1 << ": ";
        cin >> numeros[i];
        repeticiones[numeros[i]]++;
    }

    cout << "Repeticiones de cada número: " << endl;
    for (int i = 0; i < tam; i++) {
        if (repeticiones[numeros[i]] > 0) {
            cout << "El número " << numeros[i] << " se repite " << repeticiones[numeros[i]] << " vez/veces " << endl;
            repeticiones[numeros[i]] = 0;
        }
    }

    return 0;
}
```

```
Ingrese el tamaño del arreglo: 6
Ingrese el número 1: 1
Ingrese el número 2: 1
Ingrese el número 3: 2
Ingrese el número 4: 3
Ingrese el número 5: 3
Ingrese el número 6: 4
Repeticiones de cada número:
El número 1 se repite 2 vez/veces
El número 2 se repite 1 vez/veces
El número 3 se repite 2 vez/veces
El número 4 se repite 1 vez/veces
```

40. Almacene en un arreglo n números (n debe ser un número par, y como máximo 50), valide si el tamaño es par e invertir los elementos a la mitad del arreglo. Imprimir el vector original y el invertido.

Si $n=6$, $v=[1][2][3][4][5][6]$ $v(\text{invertido})=[3][2][1][6][5][4]$.

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

int main() {
    int tam;

    cout<<"Ingrese el tamaño: (Max 50) ";
    cin>>tam;

    int numeros[tam];
    for(int i=0;i<tam;i++){
        cout<<"Ingrese el número "<<i<<": ";
        cin>>numeros[i];
    }

    if (tam % 2 == 0 && tam <= 50) {
        cout<<"Arreglo original: ";
        for(int i=0;i<tam;i++){
            cout<<numeros[i]<<" ";
        }
        cout<<endl;

        int temp;
        int mitad = tam / 2;
        for (int i = 0; i < mitad/2; i++) {
            temp = numeros[i];
            numeros[i] = numeros[mitad-i-1];
            numeros[mitad - 1 - i] = temp;
        }
        cout<<"Arreglo con la primera mitad invertida: ";
        for(int i=0;i<tam;i++){
            cout<<numeros[i]<<" ";
        }

    } else {
        cout<<"El tamaño del arreglo no es válido. Debe ser un número par y como máximo 50.";
    }

    return 0;
}
```

```
Ingrese el tamaño: (Max 50) 6
Ingrese el número 0: 1
Ingrese el número 1: 2
Ingrese el número 2: 3
Ingrese el número 3: 4
Ingrese el número 4: 5
Ingrese el número 5: 6
Arreglo original: 1 2 3 4 5 6
Arreglo con la primera mitad invertida: 3 2 1 4 5 6
=====
```

41. Almacene en dos arreglos números reales a y b de 10 elementos cada uno, a partir de ellos genere un tercer arreglo con el siguiente criterio: sumar el primer elemento de a más el último elemento de b y luego el segundo elemento de a con el noveno elemento de b y así sucesivamente hasta llegar al décimo elemento de a más el primer elemento de b. Imprimir las sumas almacenadas en el arreglo c.



```
#include <iostream>

using namespace std;

int main() {
    int a[] = {31,32,34,35,36,37,38,39,40,41};
    int b[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

    int c[10];

    for (int i = 0; i < 10; i++) {
        c[i] = a[i] + b[9 - i];
    }
    cout<<"Sumas almacenadas en el arreglo c:"<<endl;
    for(int i=0;i<10;i++){
        cout<<c[i]<<" ";
    }
    return 0;
}
```

```
Sumas almacenadas en el arreglo c:
32, 34, 37, 39, 41, 43, 45, 47, 49, 51,
```

42. Almacene un arreglo de longitud n ordenado en forma ascendente y un elemento x del mismo tipo que los elementos del arreglo, intercalar x en el arreglo v de manera que siga ordenado.



```
#include <iostream>

using namespace std;

int main() {
    int n;

    cout << "Ingrese el tamaño del arreglo ordenado: ";
    cin >> n;

    int arr[100];
    for (int i = 0; i < n; i++) {
        cout << "Numero " << i << ": ";
        cin >> arr[i];
    }

    int x;
    cout << "Ingrese el número a intercalar: ";
    cin >> x;

    int i = n - 1;
    while (i >= 0 && arr[i] > x) {
        arr[i + 1] = arr[i];
        i--;
    }

    arr[i + 1] = x;
    n++;

    cout << "Arreglo después de intercalar " << x << ":\n";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

```
Ingrese el tamaño del arreglo ordenado: 5
Numero 0: 1
Numero 1: 2
Numero 2: 3
Numero 3: 4
Numero 4: 5
Ingrese el número a intercalar: 4
Arreglo después de intercalar 4:
1 2 3 4 4 5
```

43. Almacene 2 arreglos a y b de longitudes n y m, el primero ordenado en forma ascendente y el segundo ordenado de manera descendente, crear un nuevo arreglo c de n + m elementos intercalando los elementos de a y b de modo que c quede ordenado en forma ascendente.



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

void bubbleSortAsc(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void bubbleSortDesc(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] < arr[j + 1]) {
                // Intercambiar
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int b[] = {18, 16, 18, 19, 2, 4, 6, 3};
    int a[] = {3, 5, 61, 1, 9, 12, 4, 2, 9, 15};
    int c[18];
    int sum = 0;

    bubbleSortAsc(a, 10);
    bubbleSortDesc(b, 8);

    cout << "a (orden ascendente): ";
    for (int i = 0; i < 10; i++) {
        cout << a[i] << " ";
    }
    cout << endl;

    cout << "b (orden descendente): ";
    for (int i = 0; i < 8; i++) {
        cout << b[i] << " ";
    }
    cout << endl;

    int i = 0, j = 7, k = 0;
    while (i < 10 && j >= 0) {
        if (a[i] <= b[j]) {
            c[k++] = a[i++];
        } else {
            c[k++] = b[j--];
        }
    }

    cout << "c (orden ascendente): ";
    for (int i = 0; i < 18; i++) {
        cout << c[i] << " ";
    }
    cout << endl;

    return 0;
}
```

```
int main() {
    int b[] = {18, 16, 18, 19, 2, 4, 6, 3};
    int a[] = {3, 5, 61, 1, 9, 12, 4, 2, 9, 15};
    int c[18];
    int sum = 0;

    bubbleSortAsc(a, 10);
    bubbleSortDesc(b, 8);

    cout << "a (orden ascendente): ";
    for (int i = 0; i < 10; i++) {
        cout << a[i] << " ";
    }
    cout << endl;

    cout << "b (orden descendente): ";
    for (int i = 0; i < 8; i++) {
        cout << b[i] << " ";
    }
    cout << endl;

    int i = 0, j = 7, k = 0;
    while (i < 10 && j >= 0) {
        if (a[i] <= b[j]) {
            c[k++] = a[i++];
        } else {
            c[k++] = b[j--];
        }
    }

    while (i < 10) {
        c[k++] = a[i++];
    }
    while (j >= 0) {
        c[k++] = b[j--];
    }

    cout << "c (orden ascendente): ";
    for (int i = 0; i < 18; i++) {
        cout << c[i] << " ";
    }
    cout << endl;

    return 0;
}
```

Resultado:

```
a (orden ascendente): 1 2 3 4 5 9 9 12 15 61
b (orden descendente): 19 18 18 16 6 4 3 2
c (orden ascendente): 1 2 2 3 3 4 4 5 6 9 9 12 15 16 18 18 19 61
```

44. Imprimir las siguientes series:

$1 \times 9 + 2 = 11$	1
$12 \times 9 + 3 = 111$	$1 \times 1 = 1$
$123 \times 9 + 4 = 1111$	$11 \times 11 = 121$
$1234 \times 9 + 5 = 11111$	$111 \times 111 = 12321$
$12345 \times 9 + 6 = 111111$	$1111 \times 1111 = 1234321$
$123456 \times 9 + 7 = 1111111$	$11111 \times 11111 = 123454321$
$1234567 \times 9 + 8 = 11111111$	$111111 \times 111111 = 12345654321$
$12345678 \times 9 + 9 = 111111111$	$1111111 \times 1111111 = 1234567654321$
$123456789 \times 9 + 10 = 1111111111$	$11111111 \times 11111111 = 123456787654321$
	$111111111 \times 111111111 = 12345678987654321$

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

int main() {

    int num = 1;
    int edit = 2;

    cout << "Primera serie:\n";
    for (int i = 0; i < 9; i++) {
        cout << num << " * 9 + " << edit << " = " << (num * 9 + edit) << endl;
        num = num * 10 + (i + 2);
        edit++;
    }

    cout << endl;
    int n = 1;

    cout << "Segunda serie:\n";
    for (int i = 1; i < 10; i++) {
        cout << n << " * " << n << " = " << (n * n) << endl;
        n = n * 10 + 1;
    }

    return 0;
}
```

Resultado:

```
Primera serie:
1 * 9 + 2 = 11
12 * 9 + 3 = 111
123 * 9 + 4 = 1111
1234 * 9 + 5 = 11111
12345 * 9 + 6 = 111111
123456 * 9 + 7 = 1111111
1234567 * 9 + 8 = 11111111
12345678 * 9 + 9 = 111111111
123456789 * 9 + 10 = 1111111111

Segunda serie:
1 * 1 = 1
11 * 11 = 121
111 * 111 = 12321
1111 * 1111 = 1234321
11111 * 11111 = 123454321
111111 * 111111 = -539247567
1111111 * 1111111 = 1912040369
11111111 * 11111111 = -2047269199
111111111 * 111111111 = 1653732529
```

45. Calcular la suma de todos los elementos de un arreglo.



```
#include <iostream>
using namespace std;
int main() {
    int sum = 0;
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    for (int i = 0; i < 10; i++) {
        sum += a[i];
    }

    cout << "Suma de los elementos del arreglo: " << sum << endl;

    return 0;
}
```

Resultado:

Suma de los elementos del arreglo: 55

46. Realizar un arreglo que registre n palabras y muestre aquellas que inician con una vocal.

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

int main() {
    string palabras[] = {"amarillo", "hello kitty", "azul", "sopa", "chamoy"};
    char vocales[] = {'a', 'e', 'i', 'o', 'u'};
    std::string palconV[5];
    int count = 0;

    for (int i = 0; i < 5; i++) {
        char primeraLetra = tolower(palabras[i][0]);
        for (int j = 0; j < 5; j++) {
            if (primeraLetra == vocales[j]) {
                palconV[count] = palabras[i];
                count++;
                break;
            }
        }
    }

    std::cout << "Palabras que inician con una vocal: ";
    for (int i = 0; i < count; i++) {
        std::cout << palconV[i] << " ";
    }

    return 0;
}
```

Resultado:

Palabras que inician con una vocal: amarillo azul



47. Una aplicación de las computadoras es dibujar gráficos de barra (llamados histogramas). Escriba un programa que dibuje un gráfico como el siguiente donde la cantidad de asteriscos de cada línea corresponda al número que está a la par.

```
3. | ***
   |
5. | *****
   |
7. | *****
   |
9. | *****
   |_____
```

```
#include <iostream>

using namespace std;

int main() {
    int numeros[] = {3, 5, 4, 7, 9};

    for (int i = 0; i < 5; i++) {
        cout << numeros[i] << " | ";
        for (int j = 0; j < numeros[i]; j++) {
            cout << "*";
        }
        cout << endl;
    }

    cout << " ";
    for (int i = 0; i < 10; i++) { |
        cout << "_";
    }
    cout << endl;

    return 0;
}
```

Resultado:

```
3 | ***
5 | *****
4 | ****
7 | *****
9 | *****
  |_____
```

48. Escribir un programa que visualice la siguiente salida:

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```



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

int main() {
    int n = 4;

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }

    for (int i = n - 1; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }

    return 0;
}
```

Resultado:

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

49. Escriba un programa que imprima la forma en diamante siguiente.

```

      *
    * *
  * * * *
* * * * *
* * * * *
  * * * *
    * *
      *
```

```
#include <iostream>
using namespace std;
int main() {
    int n = 5;

    for (int i = 1; i <= n; i++) {
        for (int j = i; j < n; j++) {
            cout << " ";
        }
        for (int j = 1; j <= (2 * i - 2); j++) {
            cout << "* ";
        }
        cout << "*" << endl;
    }

    for (int i = n - 1; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            cout << " ";
        }
        for (int j = 1; j <= (2 * i - 2); j++) {
            cout << "* ";
        }
        cout << "*" << endl;
    }

    return 0;
}
```

Resultado:

```

      *
    * * *
  * * * * *
* * * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

50. Escriba un programa que imprima los siguientes patrones por separado, uno debajo del siguiente. Utilice ciclos "for" para generar los patrones.

(A)

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

(B)

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

(C)

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

(D)

```

      *
    **
  ***
****
*****
*****
*****
*****
*****
*****
```



```
#include <iostream>
using namespace std;
int main() {
    int n = 16;
    char arr[16];

    for (int i = 1; i <= n; i++) {
        for (int j = 6; j < i; j++) {
            arr[j] = '+';
        }
        for (int j = 6; j < i; j++) {
            cout << arr[j];
        }
        cout << endl;
    }

    cout << endl;

    for (int i = n; i >= 1; i--) {
        for (int j = 6; j < i; j++) {
            arr[j] = '+';
        }
        for (int j = 6; j < i; j++) {
            cout << arr[j];
        }
        cout << endl;
    }

    cout << endl;

    for (int i = n; i >= 1; i--) {
        for (int j = 6; j < n - i; j++) {
            cout << " ";
        }
        for (int j = 6; j < i; j++) {
            arr[j] = '+';
        }
        for (int j = 6; j < i; j++) {
            cout << arr[j];
        }
        cout << endl;
    }

    cout << endl;

    for (int i = 1; i <= n; i++) {
        for (int j = 6; j < n - i; j++) {
            cout << " ";
        }
        for (int j = 6; j < i; j++) {
            arr[j] = '+';
        }
        for (int j = 6; j < i; j++) {
            cout << arr[j];
        }
        cout << endl;
    }

    return 0;
}
```

Resultado:



51. Un programa que almacene el nombre de 15 alumnos y su número favorito, al final que imprima todos los alumnos con su número favorito, y que aparte imprima los número de menor a mayor.



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

int main() {
    int x = 15;
    string alumnos[x];
    int numeros[x];

    for (int i = 0; i < x; i++) {
        cout << "Ingrese el nombre del alumno " << i + 1 << ": ";
        cin >> alumnos[i];

        cout << "Ingrese el numero favorito del alumno " << alumnos[i] << ": ";
        cin >> numeros[i];
    }

    for (int i = 0; i < x - 1; i++) {
        for (int j = 0; j < x - i - 1; j++) {
            if (numeros[j] > numeros[j + 1]) {
                int temp = numeros[j];
                numeros[j] = numeros[j + 1];
                numeros[j + 1] = temp;

                string tempNombre = alumnos[j];
                alumnos[j] = alumnos[j + 1];
                alumnos[j + 1] = tempNombre;
            }
        }
    }

    cout << "Lista de numeros favoritos ordenados de menor a mayor con sus respectivos alumnos:" << endl;
    for (int i = 0; i < x; i++) {
        cout << alumnos[i] << ": " << numeros[i];
        if (i < x - 1) {
            cout << ", ";
        }
    }
    cout << endl;

    return 0;
}
```

Resultado:

```
Ingrese el numero favorito del alumno Jareth: 98
Ingrese el nombre del alumno 5: Cristopher
Ingrese el numero favorito del alumno Cristopher: 67
Ingrese el nombre del alumno 6: Enrique
Ingrese el numero favorito del alumno Enrique: 19
Ingrese el nombre del alumno 7: Scarlet
Ingrese el numero favorito del alumno Scarlet: 6
Ingrese el nombre del alumno 8: Monse
Ingrese el numero favorito del alumno Monse: 56
Ingrese el nombre del alumno 9: Jorge
Ingrese el numero favorito del alumno Jorge: 12
Ingrese el nombre del alumno 10: Yordi
Ingrese el numero favorito del alumno Yordi: 765
Ingrese el nombre del alumno 11: Gerardo
Ingrese el numero favorito del alumno Gerardo: 95
Ingrese el nombre del alumno 12: Wendy
Ingrese el numero favorito del alumno Wendy: 34
Ingrese el nombre del alumno 13: Daniela
Ingrese el numero favorito del alumno Daniela: 32
Ingrese el nombre del alumno 14: Maricela
Ingrese el numero favorito del alumno Maricela: 45
Ingrese el nombre del alumno 15: Xitlally
Ingrese el numero favorito del alumno Xitlally: 15
Lista de numeros favoritos ordenados de menor a mayor con sus respectivos alumnos:
tona: 1, Scarlet: 6, andres: 7, si: 8, Jorge: 12, Xitlally: 15, Enrique: 19, Daniela: 32, Wendy: 34,
Maricela: 45, Monse: 56, Cristopher: 67, Gerardo: 95, Jareth: 98, Yordi: 765
```



52. Una suma de 2 arreglos, validar que no tenga espacios vacíos.

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

int main() {
    const int size = 5;
    int arr1[size];
    int arr2[size];
    int result;

    cout << "Ingrese " << size << " elementos para el primer arreglo:" << endl;
    for (int i = 0; i < size; i++) {
        cout << "Elemento " << i + 1 << ": ";
        cin >> arr1[i];

        if (arr1[i] == 0) {
            cout << "Error: no puede ingresar cero como un valor." << endl;
            return 1;
        }
    }

    cout << "Ingrese " << size << " elementos para el segundo arreglo:" << endl;
    for (int i = 0; i < size; i++) {
        cout << "Elemento " << i + 1 << ": ";
        cin >> arr2[i];

        if (arr2[i] == 0) {
            cout << "Error: no puede ingresar cero como un valor." << endl;
            return 1;
        }
    }

    for (int i = 0; i < size; i++) {
        result += arr1[i] + arr2[i];
    }

    cout << "Resultado de la suma de los dos arreglos:" << result << endl;
}
```

Resultado:

```
Ingrese 5 elementos para el primer arreglo:
Elemento 1: 2
Elemento 2: 2
Elemento 3: 2
Elemento 4: 2
Elemento 5: 2
Ingrese 5 elementos para el segundo arreglo:
Elemento 1: 3
Elemento 2: 3
Elemento 3: 3
Elemento 4: 3
Elemento 5: 3
Resultado de la suma de los dos arreglos:25
```



53. Realizar un arreglo a partir de un número "x" donde se den los múltiplos de "n" número hasta "y" número.

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

int main() {
    int n, x, y;

    cout << "Ingrese el número para la tabla de multiplicar (n): ";
    cin >> n;
    cout << "Ingrese el valor inicial del múltiplo (x): ";
    cin >> x;
    cout << "Ingrese el valor final del múltiplo (y): ";
    cin >> y;

    if (x > y) {
        cout << "Error: el valor inicial (x) debe ser menor o igual al valor final (y)." << endl;
        return 1;
    }

    cout << "Tabla de multiplicar del " << n << " desde " << x << " hasta " << y << ":" << endl;
    for (int i = x; i <= y; i++) {
        cout << n << " x " << i << " = " << n * i << endl;
    }

    return 0;
}
```

Resultado:

```
Ingrese el n-mero para la tabla de multiplicar (n): 5
Ingrese el valor inicial del m-ltiplo (x): 3
Ingrese el valor final del m-ltiplo (y): 8
Tabla de multiplicar del 5 desde 3 hasta 8:
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
```

54. Pide al usuario que ingrese 5 nombres y almacenalos en un arreglo, luego muéstralos uno por uno.

```
#include <iostream>
using namespace std;
int main()
{
    string nombres[5];

    for (int i=0; i<5; i++) {
        cout<<"Ingrese el nombre de la persona "<<i<<":"<<endl;
        cin>>nombres[i];
        cout<<endl;
    }

    string resultado = "Los nombres ingresados son: ";
    for (int i=0; i<5; i++) {
        cout<<resultado<<i<<".- "<<nombres[i]<<endl<<endl;
    }
    return 0;
}
```



resultado

```
Ingrese el nombre de la persona 0:  
tona  
  
Ingrese el nombre de la persona 1:  
jareth  
  
Ingrese el nombre de la persona 2:  
cris  
  
Ingrese el nombre de la persona 3:  
enrique  
  
Ingrese el nombre de la persona 4:  
jorge  
  
Los nombres ingresados son: 0.- tona  
  
Los nombres ingresados son: 1.- jareth  
  
Los nombres ingresados son: 2.- cris  
  
Los nombres ingresados son: 3.- enrique  
  
Los nombres ingresados son: 4.- jorge
```

55. A partir de un arreglo de n numero de valores, contar cuantos números del arreglo son mayores al valor en la posición.


```
#include <iostream>
using namespace std;
int main()
{
    int x=5;
    cout<<"Tamaño del arreglo: "<<x<<endl;
    int numeros [5];

    for (int i=0; i<x; i++) {
        cout<<"Ingrese el valor de la posicion "<<i<<":"<<endl;
        cin>>numeros[i];
    }

    int posicion=0;
    cout<<"Ingrese la posicion del valor que quiere comparar: ";
    cin>>posicion;
    if (posicion<0 || posicion>=x) {
        cout<<"Invalido";
    }

    int valorPosicion=numeros[posicion];

    int contador=1;
    for (int i=0; i<x; i++) {
        if (numeros[i]>valorPosicion){
            contador++;
        }
    }
    cout<<"Hay "<<contador<<" numeros mayores que el valor seleccionado";
    return 0;
}
```

Resultado:

```
Ingrese el valor de la posicion 0:
4
Ingrese el valor de la posicion 1:
5
Ingrese el valor de la posicion 2:
6
Ingrese el valor de la posicion 3:
7
Ingrese el valor de la posicion 4:
8
Ingrese la posicion del valor que quiere comparar: 3
Hay 2 numeros mayores que el valor seleccionado
```

56. Realizar un arreglo que pida el nombre de n personas y ordenarlos alfabéticamente.



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

int main() {
    int n;
    cout << "Ingrese la cantidad de personas: ";
    cin >> n;

    if (n <= 0) {
        cout << "la cantidad de personas debe ser mayor que cero." << endl;
        return 1;
    }

    string personas[n];

    for (int i = 0; i < n; i++) {
        cout << "Ingrese el nombre de la persona " << i + 1 << ": ";
        cin >> personas[i];
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (personas[j] > personas[j + 1]) {
                string temp = personas[j];
                personas[j] = personas[j + 1];
                personas[j + 1] = temp;
            }
        }
    }

    cout << "Nombres ordenados alfabéticamente:" << endl;
    for (int i = 0; i < n; i++) {
        cout << personas[i] << endl;
    }

    return 0;
}
```

Resultado:

```
Ingrese la cantidad de personas: 4
Ingrese el nombre de la persona 1: Tona
Ingrese el nombre de la persona 2: Jareth
Ingrese el nombre de la persona 3: Enrique
Ingrese el nombre de la persona 4: Cris
Nombres ordenados alfabéticamente:
Cris
Enrique
Jareth
Tona
```

57. Escribe un programa que reciba un arreglo de números enteros y determine cuál es el valor máximo del arreglo.



```
#include <iostream>
using namespace std;
int main()
{
    int x=2;
    cout<<"La cantidad de personas es de: "<<x;
    cout<<endl<<endl;

    string nombres [2];
    double estaturas [2];

    for (int i=0; i<x; i++) {
        cout<<"Ingrese el nombre de la persona "<<i<<": ";
        cin>>nombres[i];
        cout<<endl;
        cout<<"Ingrese la estatura de "<<nombres[i]<<" en centimetros: ";
        cin>>estaturas[i];
        cout<<endl;
    }
    cout<<"personas registradas:"<<endl;
    for (int i=0; i<x; i++) {
        cout<<nombres[i]<<endl;
    }
    cout<<endl;

    string buscar="";
    cout<<"Ingrese el nombre de la persona para consultar su estatura: ";
    cin>>buscar;

    int i=0;
    for (i=0; i<x; i++) {
        if (nombres[i]==buscar) {
            cout<<"La estatura de "<<nombres[i]<<" es de "<<estaturas[i]<<" centimetros";
        }else{
        }
    }
    return 0;
}
```

Resultado:

```
Tamaño del arreglo: 5
Ingrese el valor de la posicion 0:5
Ingrese el valor de la posicion 1:4
Ingrese el valor de la posicion 2:7
Ingrese el valor de la posicion 3:9
Ingrese el valor de la posicion 4:4

El valor maximo en el arreglo es: 9
```

58. Realizar un programa que te pida una rima, el programa imprimirá el largo de la rima, en caso de que las palabras rimen imprimirlas, en caso de que no rimen salir un mensaje de alerta.



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

int main() {
    string rima1, rima2;

    cout << "Ingrese la primera rima: ";
    getline(cin, rima1);
    cout << "Ingrese la segunda rima: ";
    getline(cin, rima2);

    string ult3_rima1 = rima1.substr(rima1.length() - 3);
    string ult3_rima2 = rima2.substr(rima2.length() - 3);

    if (ult3_rima1 == ult3_rima2) {
        cout << "Las palabras riman :D" << endl;
    } else {
        cout << "Las palabras no riman" << endl;
    }

    return 0;
}
```

Resultado:

```
Ingrese la primera rima: Cancion
Ingrese la segunda rima: Camion
Las palabras riman :D
```

59. Un problema el cual almacene el nombre y estatura de una personas, al final que imprima solo el nombre de las personas y que el usuario pueda solicitar la estatura de cualquier persona por medio de su nombre.



```
#include <iostream>
using namespace std;
int main()
{
    int x=2;
    cout<<"La cantidad de personas es de: "<<x;
    cout<<endl<<endl;

    string nombres [2];
    double estaturas [2];

    for (int i=0; i<x; i++) {
        cout<<"Ingrese el nombre de la persona "<<i<<": ";
        cin>>nombres[i];
        cout<<endl;
        cout<<"Ingrese la estatura de "<<nombres[i]<<" en centimetros: ";
        cin>>estaturas[i];
        cout<<endl;
    }
    cout<<"personas registradas:"<<endl;
    for (int i=0; i<x; i++) {
        cout<<nombres[i]<<endl;
    }
    cout<<endl;

    string buscar="";
    cout<<"Ingrese el nombre de la persona para consultar su estatura: ";
    cin>>buscar;

    int i=0;
    for (i=0; i<x; i++) {
        if (nombres[i]==buscar) {
            cout<<"La estatura de "<<nombres[i]<<" es de "<<estaturas[i]<<" centimetros";
        }else{
        }
    }
    return 0;
}
```

Resultado:

```
La cantidad de personas es de: 2

Ingrese el nombre de la persona 0: Tona

Ingrese la estatura de Tona en centimetros: 1.56

Ingrese el nombre de la persona 1: Enrique

Ingrese la estatura de Enrique en centimetros: 1.93

personas registradas:
Tona
Enrique

Ingrese el nombre de la persona para consultar su estatura: Tona
La estatura de Tona es de 1.56 centimetros
```

60. Realizar un programa que permita ingresar los géneros de videojuegos y que los ordene, solicitando ingresar un número determinado de genero de videojuegos y ordenarlos alfabéticamente.



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

int main() {
    int n;
    cout << "Ingrese el número de géneros de videojuegos: ";
    cin >> n;

    if (n <= 0) {
        cout << "Error: el número de géneros debe ser mayor que cero." << endl;
        return 1;
    }
    string generos[n];

    for (int i = 0; i < n; i++) {
        cout << "Ingrese el género de videojuego " << i + 1 << ": ";
        cin >> ws;
        getline(cin, generos[i]);
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (generos[j] > generos[j + 1]) {
                string temp = generos[j];
                generos[j] = generos[j + 1];
                generos[j + 1] = temp;
            }
        }
    }

    cout << "Géneros de videojuegos ordenados alfabéticamente:" << endl;
    for (int i = 0; i < n; i++) {
        cout << generos[i] << endl;
    }

    return 0;
}
```

Resultado:

```
Ingrese el número de géneros de videojuegos: 3
Ingrese el género de videojuego 1: BR
Ingrese el género de videojuego 2: terror
Ingrese el género de videojuego 3: Equipo
Géneros de videojuegos ordenados alfabéticamente:
BR
Equipo
terror
```

61. Realizar un arreglo de 10 posiciones que recopile números de 1 a 10 en árabeto y haga la conversión a números romanos.
No es necesario un orden en el registro de valores del arreglo.



```
#include <iostream>
using namespace std;
int main()
{
    int numeros [10];
    string numerosR[]= {"I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX", "X"};

    for (int i=0; i<10; i++){
        numeros[i]=i+1;
    }

    cout<<"Conversion:";
    cout<<"Arabigo -> Romano"<<endl;

    for (int i=0; i<10; i++){
        cout<<numeros[i]<<" -> "<<numerosR[i]<<endl;
    }

    return 0;
}
```

Resultado:

```
Conversion:Arabigo -> Romano
1 -> I
2 -> II
3 -> III
4 -> IV
5 -> V
6 -> VI
7 -> VII
8 -> VIII
9 -> IX
10 -> X
```

62. Realizar los arreglos necesarios para almacenar la cantidad ahorrada cada mes del año y al cierre del último mes se muestre el total de ahorro mensual y anual en pesos, euros, dólares americanos, dólares canadienses, yenes y soles.



```
#include <iostream>
using namespace std;
int main()
{
    string meses []={"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};
    double ahorros[12];

    double euro=0.043;
    double dolarA=0.05;
    double dolarC=0.068;
    double yenes=6.88;
    double soles=0.203;

    double sumArr=0;
    for(int i=0; i<12; i++){
        cout<<"Ingrese el ahorro para "<<meses[i]<<": ";
        cin>>ahorros[i];
        sumArr=sumArr+ahorros[i];
    }
    cout<<endl;

    cout<<"Ahorros en Pesos";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr<<endl;
    cout<<endl;

    cout<<"Ahorros en Euros";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*euro<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*euro<<endl;
    cout<<endl;

    cout<<"Ahorros en Dolares Americanos";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*dolarA<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*dolarA<<endl;

    cout<<"Ahorros en Euros";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*euro<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*euro<<endl;
    cout<<endl;

    cout<<"Ahorros en Dolares Americanos";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*dolarA<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*dolarA<<endl;
    cout<<endl;

    cout<<"Ahorros en Dolares Canadienses";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*dolarC<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*dolarC<<endl;
    cout<<endl;

    cout<<"nAhorros en Yenes";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*yenes<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*yenes<<endl;
    cout<<endl;

    cout<<"Ahorros en Soles";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*soles<<endl;
    }
    cout<<"Ahorro Anual: "<<sumArr*soles<<endl;
    cout<<endl;

    return 0;
}
```

Resultado:



Ingrese el ahorro para Enero: 234
Ingrese el ahorro para Febrero: 3445
Ingrese el ahorro para Marzo: 666
Ingrese el ahorro para Abril: 886
Ingrese el ahorro para Mayo: 433
Ingrese el ahorro para Junio: 788
Ingrese el ahorro para Julio: 543
Ingrese el ahorro para Agosto: 987
Ingrese el ahorro para Septiembre: 447
Ingrese el ahorro para Octubre: 335
Ingrese el ahorro para Noviembre: 897
Ingrese el ahorro para Diciembre: 569

Ahorros en PesosAhorros en el mes de Enero: 234
Ahorros en el mes de Febrero: 3445
Ahorros en el mes de Marzo: 666
Ahorros en el mes de Abril: 886
Ahorros en el mes de Mayo: 433
Ahorros en el mes de Junio: 788
Ahorros en el mes de Julio: 543
Ahorros en el mes de Agosto: 987
Ahorros en el mes de Septiembre: 447
Ahorros en el mes de Octubre: 335
Ahorros en el mes de Noviembre: 897
Ahorros en el mes de Diciembre: 569
Ahorro Anual: 10230

Ahorros en EurosAhorros en el mes de Enero: 10.062
Ahorros en el mes de Febrero: 148.135
Ahorros en el mes de Marzo: 28.638
Ahorros en el mes de Abril: 38.098
Ahorros en el mes de Mayo: 18.619
Ahorros en el mes de Junio: 33.884
Ahorros en el mes de Julio: 23.349
Ahorros en el mes de Agosto: 42.441
Ahorros en el mes de Septiembre: 19.221
Ahorros en el mes de Octubre: 14.405
Ahorros en el mes de Noviembre: 38.571
Ahorros en el mes de Diciembre: 24.467
Ahorro Anual: 439.89

Ahorros en Dolares AmericanosAhorros en el mes de Enero: 11.7
Ahorros en el mes de Febrero: 172.25
Ahorros en el mes de Marzo: 33.3
Ahorros en el mes de Abril: 44.3



```
Ahorro Anual: 511.5

Ahorros en Dolares CanadiensesAhorros en el mes de Enero: 15.912
Ahorros en el mes de Febrero: 234.26
Ahorros en el mes de Marzo: 45.288
Ahorros en el mes de Abril: 60.248
Ahorros en el mes de Mayo: 29.444
Ahorros en el mes de Junio: 53.584
Ahorros en el mes de Julio: 36.924
Ahorros en el mes de Agosto: 67.116
Ahorros en el mes de Septiembre: 30.396
Ahorros en el mes de Octubre: 22.78
Ahorros en el mes de Noviembre: 60.996
Ahorros en el mes de Diciembre: 38.692
Ahorro Anual: 695.64

nAhorros en YenesAhorros en el mes de Enero: 1609.92
Ahorros en el mes de Febrero: 23701.6
Ahorros en el mes de Marzo: 4582.08
Ahorros en el mes de Abril: 6095.68
Ahorros en el mes de Mayo: 2979.04
Ahorros en el mes de Junio: 5421.44
Ahorros en el mes de Julio: 3735.84
Ahorros en el mes de Agosto: 6790.56
Ahorros en el mes de Septiembre: 3075.36
Ahorros en el mes de Octubre: 2304.8
Ahorros en el mes de Noviembre: 6171.36
Ahorros en el mes de Diciembre: 3914.72
Ahorro Anual: 70382.4

Ahorros en SolesAhorros en el mes de Enero: 47.502
Ahorros en el mes de Febrero: 699.335
Ahorros en el mes de Marzo: 135.198
Ahorros en el mes de Abril: 179.858
Ahorros en el mes de Mayo: 87.899
Ahorros en el mes de Junio: 159.964
Ahorros en el mes de Julio: 110.229
Ahorros en el mes de Agosto: 200.361
Ahorros en el mes de Septiembre: 90.741
Ahorros en el mes de Octubre: 68.005
Ahorros en el mes de Noviembre: 182.091
Ahorros en el mes de Diciembre: 115.507
Ahorro Anual: 2076.69
```

63. Realizar un programa que almacene una palabra, la descomponga en un arreglo y muestre la cantidad de vocales y consonantes que la conforman.



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

int main() {
    string palabra;
    int vocales = 0, consonantes = 0;

    cout << "Ingrese una palabra: ";
    cin >> palabra;

    for (int i = 0; i < palabra.length(); i++) {
        char c = tolower(palabra[i]);

        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
            vocales++;
        } else if (isalpha(c)) {
            consonantes++;
        }
    }

    cout << "La palabra contiene " << vocales << " vocales y " << consonantes << " consonantes" << endl;
    return 0;
}
```

Resultado:

```
Ingrese una palabra: Sopa
La palabra contiene 2 vocales y 2 consonantes
```