



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

**Nombre:**

Scarlet Joselyne Felix Gomez  
Cesar Moises Castañeda O.zazueta  
Gael Antonio López Ponce

**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



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

```
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) * 2;
8     }
9
10    cout << "Los primeros 20 números pares son: " << endl;
11    for (int i = 0; i < 20; i++) {
12        cout << pares[i] << " ";
13    }
14    cout << endl;
15
16    return 0;
17 }
```

```
Los primeros 20 números pares son:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
-----
Process exited after 0.2561 seconds with return value 0
Presione una tecla para continuar . . .
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int al;
6     cout << "Ingrese la cantidad de estudiantes: ";
7     cin >> al;
8     double calificaciones[al];
9
10    for (int i = 0; i < al; i++) {
11        cout << "Ingrese la calificación del alumno " << i + 1 << ": ";
12        cin >> calificaciones[i];
13    }
14
15    double suma = 0;
16    for (int i = 0; i < al; i++) {
17        suma += calificaciones[i];
18    }
19
20    double media = suma / al;
21    cout << "Promedio: " << media << endl;
22    cout << "Calificaciones mayores a la media:\n";
23
24    for (int i = 0; i < al; i++) {
25        if (calificaciones[i] > media) {
26            cout << calificaciones[i] << endl;
27        }
28    }
29
30    return 0;
31 }
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Ingrese la cantidad de estudiantes: 5
Ingrese la calificacion del alumno 1: 5
Ingrese la calificacion del alumno 2: 9
Ingrese la calificacion del alumno 3: 10
Ingrese la calificacion del alumno 4: 4
Ingrese la calificacion del alumno 5: 7
Promedio: 7
Calificaciones mayores a la media:
9
10
```

---

```
Process exited after 8.873 seconds with return value 0
Presione una tecla para continuar . . .
```



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 numeros[10];
6     int maximo = -1;
7     int posicion = -1;
8
9     for (int i = 0; i < 10; i++) {
10         int numero;
11         cout << "Ingrese el numero " << i + 1 << ": ";
12         cin >> numero;
13
14         for (int j = 0; j < i; j++) {
15             if (numeros[j] == numero) {
16                 cout << "El número ya fue ingresado. Intente otro número.\n";
17                 i--;
18                 break;
19             }
20         }
21
22         if (i >= 0 && numeros[i] != numero) {
23             numeros[i] = numero;
24
25             if (numero > maximo) {
26                 maximo = numero;
27                 posicion = i;
28             }
29         }
30     }
31
32     cout << "El digito mayor es: " << maximo << " en la posicion " << posicion << endl;
33
34     return 0;
35 }
```



```
Ingrese el numero 1: 5
Ingrese el numero 2: 7
Ingrese el numero 3: 8
Ingrese el numero 4: 9
Ingrese el numero 5: 4
Ingrese el numero 6: 2
Ingrese el numero 7: 1
Ingrese el numero 8: 6
Ingrese el numero 9: 4
El n·mero ya fue ingresado. Intente otro n·mero.
Ingrese el numero 9: 7
El n·mero ya fue ingresado. Intente otro n·mero.
Ingrese el numero 9: 10
Ingrese el numero 10: 12
El digito mayor es: 12 en la posicion 9
```

---

```
Process exited after 16.33 seconds with return value 0
Presione una tecla para continuar . . .
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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     int numeros[10];
6     int maximo = -1;
7     int posicion = -1;
8
9     for (int i = 0; i < 10; i++) {
10         int numero;
11         cout << "Ingrese el numero " << i + 1 << ": ";
12         cin >> numero;
13
14         for (int j = 0; j < i; j++) {
15             if (numeros[j] == numero) {
16                 cout << "El numero ya ha sido ingresado:\n";
17                 i--;
18                 j = -1;
19             }
20         }
21
22         if (i >= 0 && numeros[i] != numero) {
23             numeros[i] = numero;
24
25             if (numero > maximo) {
26                 maximo = numero;
27                 posicion = i;
28             }
29         }
30     }
31
32     cout << "El digito mayor es: " << maximo << " en la posicion " << posicion << endl;
33
34     return 0;
35 }
```

```
Ingrese el numero 1: 1
Ingrese el numero 2: 2
Ingrese el numero 3: 3
Ingrese el numero 4: 4
Ingrese el numero 5: 5
Ingrese el numero 6: 6
Ingrese el numero 7: 7
Ingrese el numero 8: 8
Ingrese el numero 9: 9
Ingrese el numero 10: 9
El numero ya ha sido ingresado:
Ingrese el numero 10: 10
El digito mayor es: 10 en la posicion 9
```

```
-----
Process exited after 19.37 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 mostrarse en el resultado, debe mostrar todos los que más ganan.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main(){
6
7     string nombres[20];
8     int sueldos[20];
9     int i=0;
10
11    while(i<2){
12        cout<<"Ingrese el nombre del empleado "<<i<<": ";
13        cin>> nombres[i];
14
15        cout<<"Ingrese el sueldo del empleado "<<i<<": ";
16        cin>>sueldos[i];
17        i++;
18    }
19
20    int max=sueldos[0];
21    int pos=0;
22
23    vector <string> mismo_sueldo;
24
25    for(int i = 0; i < 2; i++) {
26        if(sueldos[i] > max) {
27            max = sueldos[i];
28            pos = i;
29        }
30
31    }
32
33    for(int j=0;j<2;j++){
34        if(sueldos[j]==max){
35            mismo_sueldo.push_back(nombres[j]);
36        }
37    }
38
39    for(int i=0;i<mismo_sueldo.size();i++){
40        cout<<"Nombre del empleado: "<<mismo_sueldo[i]<<endl;
41    }
42
43    cout<<"Sueldo mayor: "<<sueldos[pos];
```



```
Ingrese el nombre del empleado 0: gael
Ingrese el sueldo del empleado 0: 1000
Ingrese el nombre del empleado 1: jorge
Ingrese el sueldo del empleado 1: 100000
Nombre del empleado: jorge
Sueldo mayor: 100000
```

---

```
Process exited after 13.96 seconds with return value 0
Presione una tecla para continuar . . .
```



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 numeros[5];
6     int divisor;
7     int contador = 0;
8
9     for (int i = 0; i < 5; i++) {
10         cout << "Ingrese el número " << i + 1 << ": ";
11         cin >> numeros[i];
12     }
13
14     cout << "Ingrese el divisor: ";
15     cin >> divisor;
16
17     for (int i = 0; i < 5; i++) {
18         if (numeros[i] % divisor == 0) {
19             contador++;
20         }
21     }
22
23     cout << "Hay " << contador << " números múltiplos de " << divisor << " entre los números ingresados." << endl;
24
25     return 0;
26 }
```

```
Ingrese el número 1: 5
Ingrese el número 2: 8
Ingrese el número 3: 7
Ingrese el número 4: 4
Ingrese el número 5: 2
Ingrese el divisor: 10
Hay 0 números múltiplos de 10 entre los números ingresados.
```

```
-----
Process exited after 7.628 seconds with return value 0
Presione una tecla para continuar . . . |
```

```
Ingrese el número 1: 5
Ingrese el número 2: 8
Ingrese el número 3: 4
Ingrese el número 4: 2
Ingrese el número 5: 6
Ingrese el divisor: 2
Hay 4 números múltiplos de 2 entre los números ingresados.
```

```
-----
Process exited after 7.42 seconds with return value 0
Presione una tecla para continuar . . . |
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int N;
7     cout << "Ingrese la cantidad de numeros: ";
8     cin >> N;
9
10    vector<int> numeros(N);
11    cout << "Ingrese los números en orden: ";
12    for (int i = 0; i < N; i++) {
13        cin >> numeros[i];
14    }
15
16    vector<int> no_re(N);
17    int n = 0;
18    no_re[n++] = numeros[0];
19
20    for (int i = 1; i < N; i++) {
21        if (numeros[i] != numeros[i - 1]) {
22            no_re[n++] = numeros[i];
23        }
24    }
25
26    cout << "números sin repeticiones: ";
27    for (int i = 0; i < n; i++) {
28        cout << no_re[i] << " ";
29    }
30    cout << endl;
31
32    return 0;
33 }
```



Ingrese la cantidad de numeros: 5

Ingrese los números en orden: 7

7

7

8

9

números sin repeticiones: 7 8 9

---

Process exited after 9.806 seconds with return value 0

Presione una tecla para continuar . . .



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int A[4];
6     int B[4];
7     int contador = 0;
8
9     cout << "Ingrese 4 números para el arreglo A:" << endl;
10    for (int i = 0; i < 4; i++) {
11        cout << "A[" << i << "]": ";
12        cin >> A[i];
13    }
14
15    cout << "Ingrese 4 números para el arreglo B:" << endl;
16    for (int i = 0; i < 4; i++) {
17        cout << "B[" << i << "]": ";
18        cin >> B[i];
19    }
20
21    for (int i = 0; i < 4; i++) {
22        for (int j = 0; j < 4; j++) {
23            if (A[i] == B[j]) {
24                contador++;
25            }
26        }
27    }
28
29    cout << "Cantidad de números de A que se encuentran en B: " << contador << endl;
30
31    return 0;
32 }
```

```
Ingrese 4 números para el arreglo A:
A[0]: 8
A[1]: 4
A[2]: 7
A[3]: 5
Ingrese 4 números para el arreglo B:
B[0]: 8
B[1]: 7
B[2]: 5
B[3]: 2
Cantidad de números de A que se encuentran en B: 3
```

```
-----
Process exited after 13.48 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.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int N;
6     cout << "Ingrese una cantidad de numeros: ";
7     cin >> N;
8
9     int numeros[N];
10    int contador = 0;
11
12    for (int i = 0; i < N; i++) {
13        int numero;
14        cout << "Ingrese el número " << i + 1 << ": ";
15        cin >> numero;
16
17        int no_re = 0;
18        for (int j = 0; j < contador; j++) {
19            if (numeros[j] == numero) {
20                cout << "El número ya fue ingresado\n";
21                no_re = 1;
22                i--;
23                j = contador;
24            }
25        }
26
27        if (no_re == 0) {
28            numeros[contador] = numero;
29            contador++;
30        }
31    }
32
33    cout << "Números ingresados: ";
34    for (int i = 0; i < contador; i++) {
35        cout << numeros[i] << " ";
36    }
37    cout << endl;
38
39    return 0;
40 }
```



```
Ingrese una cantidad de numeros: 5
Ingrese el n·mero 1: 5
Ingrese el n·mero 2: 5
El n·mero ya fue ingresado
Ingrese el n·mero 2: 7
Ingrese el n·mero 3: 4
Ingrese el n·mero 4: 4
El n·mero ya fue ingresado
Ingrese el n·mero 4: 7
El n·mero ya fue ingresado
Ingrese el n·mero 4: 5
El n·mero ya fue ingresado
Ingrese el n·mero 4: 1
Ingrese el n·mero 5: 10
N·meros ingresados: 5 7 4 1 10
```

---

```
Process exited after 24.42 seconds with return value 0
Presione una tecla para continuar . . .
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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     const int N = 100;
6     int numeros[N];
7
8     for (int i = 0; i < N; i++) {
9         numeros[i] = i + 1;
10    }
11
12     cout << "numeros del 1 al 100 :" << endl;
13     for (int i = 0; i < N; i++) {
14         cout << numeros[i] << " ";
15     }
16     cout << endl;
17
18     return 0;
19 }
```

```
numeros del 1 al 100 :
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 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

-----
Process exited after 0.1955 seconds with return value 0
Presione una tecla para continuar . . .
```



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

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int numeros[100];
6
7      for (int i = 0; i < 100; i++) {
8          cout << "Ingrese el número " << i + 1 << ": ";
9          cin >> numeros[i];
10     }
11
12     for (int i = 0; i < 100 - 1; i++) {
13         for (int n = 0; j < 100 - i - 1; n++) {
14             if (numeros[n] < numeros[n + 1]) {
15
16                 int temp = numeros[n];
17                 numeros[j] = numeros[n + 1];
18                 numeros[n + 1] = temp;
19             }
20         }
21     }
22
23     cout << "Números ordenados de mayor a menor: ";
24     for (int i = 0; i < 100; i++) {
25         cout << numeros[i] << " ";
26     }
27     cout << endl;
28
29     return 0;
30 }
```

```
Ingrese el número 92: 93
Ingrese el número 93: 94
Ingrese el número 94: 95
Ingrese el número 95: 96
Ingrese el número 96: 97
Ingrese el número 97: 98
Ingrese el número 98: 99
Ingrese el número 99: 100
Ingrese el número 100: 101
Números ordenados de mayor a menor: 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 78 75 70 69
69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 50 49 48 47 47 46 45 44 43 42 41 40 40 39 38 37 36 35 34 33 32 31 30 16 14
13 12 9 8 8 8 7 7 6 6 6 5 5 5 4 3 3 2 2 1 1 0

Process exited after 159 seconds with return value 0
Presione una tecla para continuar . . .
```



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 impares[50];
6     int n = 0;
7
8     for (int i = 1; i <= 100; i += 2) {
9         impares[n++] = i;
10    }
11
12    cout << "Números impares del 1 al 100: ";
13    for (int i = 0; i < 50; i++) {
14        cout << impares[i] << " ";
15    }
16    cout << endl;
17
18    return 0;
19 }
20
```

```
Números impares del 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 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 0.1959 seconds with return value 0
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 numeros[10];
6     int suma = 0;
7     int resta = 0;
8     int multiplicacion = 1;
9     double division;
10
11    for (int i = 0; i < 10; i++) {
12        cout << "Ingrese el número " << i + 1 << ": ";
13        cin >> numeros[i];
14    }
15
16    resta = numeros[0];
17    division = numeros[0];
18
19    for (int i = 0; i < 10; i++) {
20        suma += numeros[i];
21        if (i > 0) {
22            resta -= numeros[i];
23            division /= numeros[i];
24        }
25        multiplicacion *= numeros[i];
26    }
27
28    cout << "Suma: " << suma << endl;
29    cout << "Resta: " << resta << endl;
30    cout << "Multiplicación: " << multiplicacion << endl;
31    cout << "División: " << division << endl;
32
33    return 0;
34 }
```



```
Ingrese el n·mero 1: 5
Ingrese el n·mero 2: 8
Ingrese el n·mero 3: 7
Ingrese el n·mero 4: 3
Ingrese el n·mero 5: 6
Ingrese el n·mero 6: 9
Ingrese el n·mero 7: 4
Ingrese el n·mero 8: 2
Ingrese el n·mero 9: 78
Ingrese el n·mero 10: 42
Suma: 164
Resta: -154
Multiplicaci·n: 1188794880
Divisi·n: 2.10297e-008
```

---

```
Process exited after 14.87 seconds with return value 0
Presione una tecla para continuar . . .
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int impares[50];
6     int pares = 0;
7
8     for (int i = 1; i <= 100; i += 2) {
9         impares[pares] = i;
10        pares++;
11    }
12
13    for (int i = 49; i >= 0; i--) {
14        cout << impares[i] << " ";
15    }
16
17    cout << endl;
18    return 0;
19 }
```

```
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 43 41 39 37 35 33 31 29 27 25 23 21
19 17 15 13 11 9 7 5 3 1
```

```
-----  
Process exited after 0.1574 seconds with return value 0  
Presione una tecla para continuar . . .
```



15. Realizar un programa que genere 100 números al azar y los ordene descendenteamente

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

Números ordenados descendente:

```
100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61
60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 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
```



16. Realice un programa que tome 10 números y los ordene ascendenteamente

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     const int n = 10;
6     int arreglo[n];
7
8     cout << "Introduce 10 números:\n";
9     for (int i = 0; i < n; i++) {
10        cout << "Número " << (i + 1) << ": ";
11        cin >> arreglo[i];
12    }
13
14    for (int i = 0; i < n - 1; i++) {
15        for (int j = 0; j < n - i - 1; j++) {
16            if (arreglo[j] > arreglo[j + 1]) {
17
18                int temp = arreglo[j];
19                arreglo[j] = arreglo[j + 1];
20                arreglo[j + 1] = temp;
21            }
22        }
23    }
24
25    cout << "\nLos números ordenados ascendentemente son:\n";
26    for (int i = 0; i < n; i++) {
27        cout << arreglo[i] << " ";
28    }
29    cout << endl;
30
31    return 0;
32}
33
```



Introduce 10 n̄umeros:

N̄úmero 1: 1  
N̄úmero 2: 2  
N̄úmero 3: 3  
N̄úmero 4: 4  
N̄úmero 5: 5  
N̄úmero 6: 6  
N̄úmero 7: 7  
N̄úmero 8: 10  
N̄úmero 9: 78  
N̄úmero 10: -5

Los n̄umeros ordenados ascendente mente son:

-5 1 2 3 4 5 6 7 10 78

---

Process exited after 15.58 seconds with return value 0  
Presione una tecla para continuar . . . |



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     const int n = 5;
6     int arreglo1[n];
7     int arreglo2[n];
8     int Final[2 * n];
9
10    cout << "Introduce 5 números para el primer arreglo:\n";
11    for (int i = 0; i < n; i++) {
12        cout << "Número " << (i + 1) << ": ";
13        cin >> arreglo1[i];
14    }
15
16    cout << "Introduce 5 números para el segundo arreglo:\n";
17    for (int i = 0; i < n; i++) {
18        cout << "Número " << (i + 1) << ": ";
19        cin >> arreglo2[i];
20    }
21
22    for (int i = 0; i < n; i++) {
23        Final[i] = arreglo1[i];
24        Final[i + n] = arreglo2[i];
25    }
26
27    cout << "\n son:\n";
28    for (int i = 0; i < 2 * n; i++) {
29        cout << Final[i] << " ";
30    }
31    cout << endl;
32
33    return 0;
34 }
```



```
Introduce 5 números para el primer arreglo:
```

```
Número 1: 18
```

```
Número 2: 19
```

```
Número 3: 20
```

```
Número 4: 17
```

```
Número 5: 16
```

```
Introduce 5 números para el segundo arreglo:
```

```
Número 1: 16
```

```
Número 2: 17
```

```
Número 3: 18
```

```
Número 4: 19
```

```
Número 5: 20
```

```
son:
```

```
18 19 20 17 16 16 17 18 19 20
```

---

```
-----  
Process exited after 47.12 seconds with return value 0
```

```
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 ascendenteamente usando un 3er arreglo.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     const int n = 5;
6     int arreglo1[n];
7     int arreglo2[n];
8     int arreglo3[2 * n];
9
10    cout << "Introduce 5 números:\n";
11    for (int i = 0; i < n; i++) {
12        cout << "Número " << (i + 1) << ": ";
13        cin >> arreglo1[i];
14        arreglo2[i] = arreglo1[i] * 2;
15    }
16
17    for (int i = 0; i < n; i++) {
18        arreglo3[i] = arreglo1[i];
19        arreglo3[i + n] = arreglo2[i];
20    }
21
22    for (int i = 0; i < 2 * n - 1; i++) {
23        for (int j = 0; j < 2 * n - i - 1; j++) {
24            if (arreglo3[j] > arreglo3[j + 1]) {
25
26                int temp = arreglo3[j];
27                arreglo3[j] = arreglo3[j + 1];
28                arreglo3[j + 1] = temp;
29            }
30        }
31    }
32
33    cout << "\nArreglo ordenado:\n";
34    for (int i = 0; i < 2 * n; i++) {
35        cout << arreglo3[i] << " ";
36    }
37    cout << endl;
38
39    return 0;
40 }
```



Introduce 5 números:

Número 1: 5  
Número 2: 78  
Número 3: 4  
Número 4: 6  
Número 5: 9

Arreglo ordenado:

4 5 6 8 9 10 12 18 78 156

---

Process exited after 10.47 seconds with return value 0  
Presione una tecla para continuar . . .



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int arreglo[10];
6     int temp;
7
8     cout << "Ingresa 10 números, pueden ser positivos o negativos:" << endl;
9     for (int i = 0; i < 10; i++) {
10         cout << "Número [" << i + 1 << "]: ";
11         cin >> arreglo[i];
12     }
13
14     for (int i = 0; i < 10; i++) {
15         for (int j = i + 1; j < 10; j++) {
16             if (arreglo[i] > arreglo[j]) {
17                 temp = arreglo[i];
18                 arreglo[i] = arreglo[j];
19                 arreglo[j] = temp;
20             }
21         }
22     }
23
24     cout << "Los números ordenados ascendenteamente son:" << endl;
25     for (int i = 0; i < 10; i++) {
26         cout << arreglo[i] << " ";
27     }
28     cout << endl;
29
30     return 0;
31 }
```

```
Ingresa 10 números, pueden ser positivos o negativos:
Número [1]: -9
Número [2]: -500
Número [3]: 10
Número [4]: 3
Número [5]: 9
Número [6]: 8
Número [7]: 7
Número [8]: -100
Número [9]: -6
Número [10]: -20
Los números ordenados ascendenteamente son:
-500 -100 -20 -9 -6 3 7 8 9 10
```

---

```
Process exited after 42.08 seconds with return value 0
Presione una tecla para continuar . . .
```



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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int arreglo[20];
6     int buscar_n, posicion = -1;
7
8     cout << "Ingresa 20 números para llenar el arreglo:" << endl;
9     for (int i = 0; i < 20; i++) {
10         cout << "Elemento [" << i + 1 << "]: ";
11         cin >> arreglo[i];
12     }
13
14     cout << "Ingresa el número que deseas buscar: ";
15     cin >> buscar_n;
16
17     for (int i = 0; i < 20; i++) {
18         if (arreglo[i] == buscar_n) {
19             posicion = i;
20         }
21     }
22
23     if (posicion != -1) {
24         cout << "El número " << buscar_n << " se encuentra en la posición " << posicion << " del arreglo." << endl;
25     } else {
26         cout << "El número " << buscar_n << " no se encuentra en el arreglo." << endl;
27     }
28
29     return 0;
30 }
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

Ingresá 20 números para llenar el arreglo:

```
Elemento [1]: 1
Elemento [2]: 2
Elemento [3]: 3
Elemento [4]: 4
Elemento [5]: 5
Elemento [6]: 6
Elemento [7]: 7
Elemento [8]: 8
Elemento [9]: 9
Elemento [10]: 10
Elemento [11]: 11
Elemento [12]: 12
Elemento [13]: 13
Elemento [14]: 14
Elemento [15]: 15
Elemento [16]: 16
Elemento [17]: 17
Elemento [18]: 18
Elemento [19]: 19
Elemento [20]: 20
```

Ingresá el número que deseas buscar: 17

El número 17 se encuentra en la posición 16 del arreglo.

---

```
Process exited after 206.2 seconds with return value 0
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.

```
programa21_arreglos.cpp | programa21_arreglos.cpp |
1  /*Arreglo 21*/          31   cin>>numb;
2                           32   int posicion = -1;
3   #include <iostream>      33   for (int i = 0; i < 100; i++) {
4   #include <cstdlib>       34       if (numeros[i] == numb) {
5   #include <ctime>        35           posicion = i;
6                           36           break;
7   using namespace std;     37       }
8                           38   }
9   int main() {            39   if (posicion == -1) {
10      int numeros [100];  40       cout<<"Número buscado: "<<numb<<"\nEl número no se encuentra en el arreglo."<<endl;
11      srand(time(0));    41   } else {
12      for (int i = 0; i < 100; i++) { 42       cout<<"Número buscado: "<<numb<<"\nPosición del arreglo en la que se encuentra: "<<posicion<<endl;
13          numeros[i] = rand() % 101; 43   }
14      }
15      for (int i = 0; i < 20 - 1; i++) {
16          for (int j = 0; j < 20 - i - 1; j++) {
17              if (numeros[j] > numeros[j + 1]) {
18                  int temp = numeros[j];
19                  numeros[j] = numeros[j + 1];
20                  numeros[j + 1] = temp;
21              }
22          }
23      }
24  }
25  }
26  }
27  }
28
29  int numb;
30  cout<<"Ingrese un número para buscar: ";
31  cin>>numb;
32  int posicion = -1;
33  for (int i = 0; i < 100; i++) {
34      if (numeros[i] == numb) {
35          posicion = i;
36      }
37  }
38
39  cin>>numb;
40  int posicion = -1;
41  for (int i = 0; i < 100; i++) {
42      if (numeros[i] == numb) {
43          posicion = i;
44          break;
45      }
46  }
47  if (posicion == -1) {
48      cout<<"Número buscado: "<<numb<<"\nEl número no se encuentra en el arreglo."<<endl;
49  } else {
50      cout<<"Número buscado: "<<numb<<"\nPosición del arreglo en la que se encuentra: "<<posicion<<endl;
51  }
52
53  int rep = 0;
54  for (int i = 0; i < 100; i++) {
55      if (numeros[i] == numb) {
56          rep++;
57      }
58  }
59  cout<<"Número de repeticiones: "<<rep<<endl;
60
61  int repArreglo;
62  cout<<"¿Cuántas veces desea ver el arreglo?"<<endl;
63  cin>>repArreglo;
64
65  for (int i = 1; i <= repArreglo; i++) {
66      cout<<"\nArreglo: "<<endl;
67      for(int i = 0; i < 100; i++){
68          cout<<numeros[i]<<" ";
69      }
70  }
71
72  return 0;
73 }
```

Ingrese un número para buscar: 5

Número buscado: 5

Posición del arreglo en la que se encuentra: 0

Número de repeticiones: 1

¿Cuántas veces desea ver el arreglo?

2

Arreglo:

5 18 23 24 32 32 35 37 38 40 43 56 58 60 68 72 78 90 96 98 11 85 26 88 62 47 78 83 82 54 23 51 75 79 48 12 3 15 50 32 45  
58 94 70 56 79 38 69 15 77 69 74 26 7 49 96 41 59 25 75 19 30 33 73 3 63 34 98 21 0 80 71 59 64 27 82 31 17 82 69 59 59  
99 31 50 21 58 90 86 42 34 94 59 58 68 39 98 9 77 23

Arreglo:

5 18 23 24 32 32 35 37 38 40 43 56 58 60 68 72 78 90 96 98 11 85 26 88 62 47 78 83 82 54 23 51 75 79 48 12 3 15 50 32 45  
58 94 70 56 79 38 69 15 77 69 74 26 7 49 96 41 59 25 75 19 30 33 73 3 63 34 98 21 0 80 71 59 64 27 82 31 17 82 69 59 59  
99 31 50 21 58 90 86 42 34 94 59 58 68 39 98 9 77 23



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

programa22\_arreglos.cpp

```
1  /*Arreglo 22*/
2
3  #include <iostream>
4  #include <cstdlib>
5  #include <ctime>
6  using namespace std;
7
8 int main() {
9     int cantProd;
10    cout<<"Ingrese la cantidad de productos que desea guardar: "<<endl;
11    cin>>cantProd;
12
13    string claves [cantProd];
14    string descfs [cantProd];
15    string exis [cantProd];
16    int minExis [cantProd];
17    double prcUnis [cantProd];
18
19    for (int i = 0; i < cantProd; i++) {
20        cout<<"Ingrese la clave del producto: "<<i<<": "<<endl;
21        cin>>claves[i];
22        cout<<"Ingrese la descripción del producto: "<<i<<": "<<endl;
23        cin>>descfs[i];
24        cout<<"Ingrese la existencia del producto (activa/inactiva): "<<i<<": "<<endl;
25        cin>>exis[i];
26        cout<<"Ingrese el mínimo para mantener la existencia del producto: "<<i<<": "<<endl;
27        cin>>minExis[i];
28        cout<<"Ingrese el precio del producto: "<<i<<": "<<endl;
29        cin>>prcUnis[i];
30    }
31
32    for (int i = 0; i < cantProd; i++) {
33        cout<<"Producto "<<i<<endl;
34        cout<<"Clave: "<<claves[i]<<endl;
35        cout<<"Descripción: "<<descfs[i]<<endl;
36        cout<<"Existencia: "<<exis[i]<<endl;
37        cout<<"Mínimo: "<<minExis[i]<<endl;
38        cout<<"Precio Unitario: $"<<prcUnis[i]<<endl;
39        cout<<endl;
40    }
41
42    return 0;
}
```

Ingrese el precio del producto: 0:  
18  
Ingrese la clave del producto: 1:  
P002  
Ingrese la descripción del producto: 1:  
Refresco  
Ingrese la existencia del producto (activa/inactiva): 1:  
activa  
Ingrese el mÍnimo para mantener la existencia del producto: 1:  
15  
Ingrese el precio del producto: 1:  
20  
Producto 0  
Clave: P001  
Descripción: Galletas  
Existencia: activa  
MÍnimo: 10  
Precio Unitario: \$18  
  
Producto 1  
Clave: P002  
Descripción: Refresco  
Existencia: activa  
MÍnimo: 15  
Precio Unitario: \$20



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

```
programa23_arreglos.cpp
1 /*Arreglo_23*/
2
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 int main() {
8
9     int productos [5] = {1, 2, 3, 4, 5};
10    int claves [5] = {1001, 1002, 1003, 1004, 1005};
11    double precios [5] = {50, 250, 200, 100, 150};
12    int stocks [5] = {50, 25, 30, 45, 50};
13    int stMin [5] = {10, 5, 7, 9, 10};
14
15    int op;
16
17    do {
18        cout<<"Seleccione una opción:\nVenta (1)\nReabastecimiento (2)\nActualizar precio (3)\nInformar (4)\nCerrar (5)"<<endl;
19        cin>>op;
20
21        if (op == 1) {
22            int clv;
23            int cant;
24            cout<<"Ingrese la clave del producto: "<<endl;
25            cin>>clv;
26            cout<<"Ingrese la cantidad a vender: "<<endl;
27            cin>>cant;
28            for (int i = 0; i < 5; i++) {
29                if (claves[i] == clv) {
30                    if (stocks[i] - cant >= stMin[i]) {
31                        stocks[i] = stocks[i] - cant;
32                        cout<<"Venta\nSe ha realizado una venta del producto con clave: "<<clv<<"\nStock restante: "<<stocks[i]<<endl;
33                    } else {
34                        cout<<"No se puede realizar la venta del producto con clave: "<<clv<<", el stock es insuficiente"<<endl;
35                }
36            }
37        }
38    } else if (op == 2) {
39        int clv;
40        int cant;
41        cout<<"Ingrese la clave del producto: "<<endl;
42        cin>>clv;
43        cout<<"Ingrese la cantidad a reabastecer: "<<endl;
44        cin>>cant;
45
46        for (int i = 0; i < 5; i++) {
47            if (claves[i] == clv) {
48                stocks[i] = stocks[i] + cant;
49                cout<<"Reabastecimiento\nSe ha realizado un reabastecimiento del producto con clave: "<<clv<<"\nNuevo stock: "<<stocks[i]<<endl;
50            }
51        }
52    } else if (op == 3) {
53        int clv;
54        double porj;
55        cout<<"Ingrese la clave del producto: "<<endl;
56        cin>>clv;
57        cout<<"Ingrese el porcentaje de aumento: "<<endl;
58        cin >> porj;
59
60        for (int i = 0; i < 5; i++) {
61            if (claves[i] == clv) {
62                precios[i] += precios[i] * (porj / 100);
63                cout<<fixed<<setprecision(2)<<"Actualización\nSe ha actualizado el precio del producto con clave: "<<
64                clv<<"\nNuevo precio: $"<<precios[i]<<endl;
65            }
66        }
67    } else if (op == 4) {
68        int clv;
69        cout << "Ingrese la clave del producto: ";
70        cin >> clv;
```



```
71
72
73     for (int i = 0; i < 5; i++) {
74         if (claves[i] == clv) {cout<<"Información"
75             <<"\nClave: "<<claves[i]
76             <<"\nNombre: "<<productos[i]
77             <<"\nPrecio: $"<<fixed<<setprecision(2)<<precios[i]
78             <<"\nStock: "<<stocks[i]
79             <<"\nStock mÃnimo: "<<stMin[i]<<endl;
80     }
81 } else if (op != 5) {
82     cout<<"La opción ingresada no es válida"<<endl;
83 }
84 } while (op != 5);
85
86 cout << "Fin";
87 return 0;
88 }
```

```
Seleccione una opción:
Venta (1)
Reabastecimiento (2)
Actualizar precio (3)
Informar (4)
Cerrar (5)
1
Ingrese la clave del producto:
1001
Ingrese la cantidad a vender:
10
Venta
Se ha realizado una venta del producto con clave: 1001
Stock restante: 40
```

```
Seleccione una opción:
Venta (1)
Reabastecimiento (2)
Actualizar precio (3)
Informar (4)
Cerrar (5)
2
Ingrese la clave del producto:
1002
Ingrese la cantidad a reabastecer:
20
Reabastecimiento
Se ha realizado un reabastecimiento del producto con clave: 1002
Nuevo stock: 45
```

```
Seleccione una opción:
Venta (1)
Reabastecimiento (2)
Actualizar precio (3)
Informar (4)
Cerrar (5)
3
Ingrese la clave del producto:
1003
Ingrese el porcentaje de aumento:
25
Actualización
Se a actualizado el precio del producto con clave: 1003
Nuevo precio: $250.00
```

```
Seleccione una opción:
Venta (1)
Reabastecimiento (2)
Actualizar precio (3)
Informar (4)
Cerrar (5)
4
Ingrese la clave del producto: 1004
Información
Clave: 1004
Nombre: 4
Precio: $100.00
Stock: 45
Stock mínimo: 9
```



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.

<pre>programa24_arreglos.cpp 1 /*Arreglo 24*/ 2 3 #include &lt;iostream&gt; 4 #include &lt;vector&gt; 5 #include &lt;iomanip&gt; 6 7 using namespace std; 8 9 int main() { 10     int numPac; 11     cout&lt;&lt;"Ingrese la cantidad de pacientes a registrar: "&lt;&lt;endl; 12     cin&gt;&gt;numPac; 13 14     string nombres [numPac]; 15     int edades [numPac]; 16     string sexos [numPac]; 17     string calles [numPac]; 18     string numeros [numPac]; 19     string ciudades [numPac]; 20     string telefonos [numPac]; 21     string seguros [numPac]; 22 23     for (int i = 0; i &lt; numPac; i++) { 24         cout&lt;&lt;"Ingrese el nombre del paciente "&lt;&lt;i&lt;&lt; ": "&lt;&lt;endl; 25         cin&gt;&gt;nombres[i]; 26         cout&lt;&lt;"Ingrese la edad del paciente " &lt;&lt;i&lt;&lt; ": "&lt;&lt;endl; 27         cin&gt;&gt;edades[i]; 28         cout&lt;&lt;"Ingrese el sexo del paciente " &lt;&lt;i&lt;&lt;" (M/F): "&lt;&lt;endl; 29         cin&gt;&gt;sexos[i]; 30         cout&lt;&lt;"Ingrese la calle del domicilio del paciente "&lt;&lt;i&lt;&lt;": "&lt;&lt;endl; 31         cin&gt;&gt;calles[i]; 32         cout&lt;&lt;"Ingrese el número de domicilio del paciente "&lt;&lt;i&lt;&lt;": "&lt;&lt;endl; 33         cin&gt;&gt;numeros[i]; 34         cout&lt;&lt;"Ingrese la ciudad del paciente "&lt;&lt;i&lt;&lt; ": "&lt;&lt;endl; 35         cin&gt;&gt;ciudades[i];</pre>	<pre>programa24_arreglos.cpp 36 37     cout&lt;&lt;"¿Tiene seguro? (Si/No): "; 38     cin&gt;&gt;seguros[i]; 39 40     string op; 41     do { 42         cout&lt;&lt;"Seleccione una opción:" 43         &lt;&lt;"\nLista de pacientes hospitalizados (a)" 44         &lt;&lt;"\nPorcentaje de pacientes hospitalizados (por edad) (b)" 45         &lt;&lt;"\nPorcentaje de pacientes hospitalizados (por sexo) (c)" 46         &lt;&lt;"\nObtener datos de un paciente (d)" 47         &lt;&lt;"\nCerrar (e)"&lt;&lt;endl; 48     cin&gt;&gt;op; 49 50     if (op == "a") { 51         cout&lt;&lt;"\nLista de pacientes hospitalizados:"&lt;&lt;endl; 52         for (int i = 0; i &lt; numPac; i++) { 53             cout&lt;&lt;i&lt;&lt;. "&lt;&lt;nombres[i]&lt;&lt;endl; 54         } 55         cout&lt;&lt;endl; 56 57     } else if (op == "b") { 58         int n = 0; 59         int j = 0; 60         int a = 0; 61         for (int i = 0; i &lt; numPac; i++) { 62             if (edades[i] &lt;= 13){ 63                 n++; 64             } else if (edades[i] &lt; 30){ 65                 j++; 66             } else { 67                 a++; 68             } 69         } 70         int porN = (n / numPac) * 100;</pre>
---	--



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
programa24_arreglos.cpp
71   int porJ = (j / numPac) * 100;
72   int porA = (a / numPac) * 100;
73   cout<<"\nPorcentaje de pacientes hospitalizados (por edad)"
74     <<"\nPorcentaje de pacientes niños: "<<porN<< "%"
75     <<"\nPorcentaje de pacientes jóvenes: "<<porJ<< "%"
76     <<"\nPorcentaje de pacientes adultos: "<<porA<< "%\n"<<endl;
77
78 } else if (op == "c") {
79   int h = 0;
80   int m = 0;
81   for (int i = 0; i < numPac; i++) {
82     if (sexos[i] == "M"){
83       h++;
84     }
85     else if (sexos[i] == "F"){
86       m++;
87     }
88   }
89   int porH = (h / numPac) * 100;
90   int porM = (m / numPac) * 100;
91   cout <<"\nPorcentaje de pacientes hospitalizados (por sexo)"
92     <<"\nPorcentaje de pacientes hombres: "<<porH<< "%"
93     <<"\nPorcentaje de pacientes mujeres: "<<porM<< "%\n"<<endl;
94
95 } else if (op == "d") {
96   string nomB;
97   cout<<"Ingrese el nombre del paciente a buscar: "<<endl;
98   cin>>nomB;
99
100  int encontrado = 0;
101  for (int i = 0; i < numPac; i++) {
102    if (nombres[i] == nomB) {
103      cout<<"\nInformación del paciente:"
104        <<"\nNombre: "<<nombres[i]
105        <<"\nEdad: "<<edades[i]
```

Seleccione una opción:  
Lista de pacientes hospitalizados (a)  
Porcentaje de pacientes hospitalizados (por edad) (b)  
Porcentaje de pacientes hospitalizados (por sexo) (c)  
Obtener datos de un paciente (d)  
Cerrar (e)

a

Lista de pacientes hospitalizados:

0. Maria  
1. Pablo

Seleccione una opción:  
Lista de pacientes hospitalizados (a)  
Porcentaje de pacientes hospitalizados (por edad) (b)  
Porcentaje de pacientes hospitalizados (por sexo) (c)  
Obtener datos de un paciente (d)  
Cerrar (e)

d

Ingrese el nombre del paciente a buscar:  
Maria

Información del paciente:  
Nombre: Maria  
Edad: 15  
Sexo: F  
Domicilio:  
Calle: morelos  
Número: 25  
Ciudad: Rosario  
Seguro: si

```
106   <<"\nSexo: "<<sexos[i]
107   <<"\nDomicilio: \nCalle: "<<calles[i]<<"\nNúmero: "<<numeros[i]<<"\nCiudad: "<<ciudades[i]
108   <<"\nSeguro: "<<seguros[i]<<"\n";
109   encontrado = 1;
110   break;
111 }
112 if (encontrado == 0) {
113   cout<<"Paciente no encontrado\n";
114 }
115
116 } else if (op != "e") {
117   cout<<"Esta opción no es válida. Intente de nuevo.\n";
118 }
119
120 } while (op != "e");
121
122 cout<<"Fin";
123
124 return 0;
125
126 }
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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:

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:

- El número de votos de cada candidato al final de la elección.
- 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.

programa25\_arreglos.cpp

```
1  /*Ejercicio 25*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7      int votos [12] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
8      int voto;
9      int ttVts = 0;
10
11     do {
12         cout<<"Ingrese el número del candidato (1-12) para registrar un voto (para detener ingrese un número negativo): ";
13         cin>>voto;
14         if (voto >= 1 && voto <= 12) {
15             votos[voto - 1]++;
16             ttVts++;
17         }
18     }while (voto >= 0);
19
20     for (int i = 0; i < 12; i++){
21         cout<<"Total de votos para el candidato "<<(i+1)<<": "<<votos[i]<<" votos"<<endl;
22     }
23
24     int maxVts = votos[0];
25     int candG = 1;
26     for (int i = 1; i < 12; i++) {
27         if (votos[i] > maxVts) {
28             maxVts = votos[i];
29             candG = i + 1;
30         }
31     }
32     double porcG = (maxVts / (double) ttVts) * 100;
33
34     cout<<"Candidato ganador: "<<candG<<endl;
35     cout<<"Número de votos: "<<maxVts<<endl;
36     cout<<"Porcentaje del total de los votos: "<<porcG<<"%"<<endl;
```

```
Total de votos para el candidato 1: 3 votos
Total de votos para el candidato 2: 1 votos
Total de votos para el candidato 3: 1 votos
Total de votos para el candidato 4: 1 votos
Total de votos para el candidato 5: 1 votos
Total de votos para el candidato 6: 1 votos
Total de votos para el candidato 7: 1 votos
Total de votos para el candidato 8: 1 votos
Total de votos para el candidato 9: 1 votos
Total de votos para el candidato 10: 1 votos
Total de votos para el candidato 11: 1 votos
Total de votos para el candidato 12: 1 votos
Candidato ganador: 1
Número de votos: 3
Porcentaje del total de los votos: 21.4286%
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

programa26\_arreglos.cpp

```
1  /*Ejercicio 26*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7      int temperaturas [24] = {26, 26, 25, 25, 25, 24, 24, 23, 24, 26, 29, 31, 32, 32, 33, 32, 31, 31, 31, 30, 28, 28, 27, 26, 26, 26};
8
9      int suma = 0;
10
11     int max = temperaturas[0];
12     int posMax = 0;
13
14     int min = temperaturas[0];
15     int posMin = 0;
16
17     for (int i = 1; i < 24; i++) {
18         suma = suma + temperaturas[i];
19         if (temperaturas[i] > max) {
20             max = temperaturas[i];
21             posMax = i;
22         }
23         if (temperaturas[i] < min) {
24             min = temperaturas[i];
25             posMin = i;
26         }
27     }
28     double media = suma / 24;
29
30     cout<<"Temperaturas: "<<endl;
31     for(int i = 0; i < 24; i++){
32         cout<<temperaturas[i]<<" ";
33     }
34
35     cout <<"\nMedia de las temperaturas: "<<media<<"°C"<<endl;
36     cout <<"Temperatura máxima: "<<max<<"°C a las "<<posMax<<" horas"<<endl;
37     cout <<"Temperatura mínima: "<<min<<"°C a las "<<posMin<<" horas"<<endl;
38
39     return 0;
40 }
```

```
Temperaturas:
26 26 25 25 24 24 23 24 26 29 31 32 32 33 32 31 31 30 28 28 27 26 26 26
Media de las temperaturas: 26°C
Temperatura máxima: 33°C a las 13 horas
Temperatura mínima: 23°C a las 6 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.

```
programa27_arreglos.cpp
1  /*Arreglo 27*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7      int temperaturas [7] = {29, 30, 30, 32, 31, 33, 31};
8      double diferencias [7];
9
10     int suma = 0;
11
12     int max = temperaturas[0];
13     int diaMax = 0;
14
15     int min = temperaturas[0];
16     int diaMin = 0;
17
18     for (int i = 0; i < 7; i++) {
19         suma = suma + temperaturas[i];
20     }
21     double media = suma / 7;
22
23     for (int i = 0; i < 7; i++) {
24         diferencias[i] = (temperaturas[i] - media);
25         if (temperaturas[i] > max) {
26             max = temperaturas[i];
27             diaMax = i;
28         }if (temperaturas[i] < min){
29             min = temperaturas[i];
30             diaMin = i;
31         }
32     }
33     cout<<"Temperaturas: "<<endl;
34     for(int i = 0; i < 7; i++){
35         cout<<temperaturas[i]<<" ";
36     }
37
38     cout<<"\nTemperatura promedio: "<<media<<"°C"<<endl;
39
40     cout << "Diferencias de las temperaturas con respecto al promedio: "<<endl;
41     for (int i = 0; i < 7; i++) {
42         cout<<"Día "<<(i + 1)<<": "<<diferencias[i]<<"°C"<<endl;
43     }
44
45     cout<<"\nTemperatura mínima: "<<min<<"°C el día "<<diaMin<<endl;
46     cout<<"Temperatura máxima: "<<max<<"°C el día "<<diaMax<<endl;
47
48 }
```

```
Temperaturas:
29 30 30 32 31 33 31
Temperatura promedio: 30°C
Diferencias de las temperaturas con respecto al promedio:
Día 1: -1°C
Día 2: 0°C
Día 3: 0°C
Día 4: 2°C
Día 5: 1°C
Día 6: 3°C
Día 7: 1°C

Temperatura mínima: 29°C el día 0
Temperatura máxima: 33°C el día 5
```



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.

programa28\_arreglos.cpp

```
1  /*Ejercicio 28*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7
8      int numeros [20] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20};
9
10     int suma = 0;
11     int maxImp = 0;
12     int maxPar = 0;
13
14     for (int i = 0; i < 20; i++) {
15         if (i % 2 == 0) {
16             suma = suma + numeros[i];
17             if (numeros[i] > maxPar) {
18                 maxPar = numeros[i];
19             }
20             if(i % 2 != 0){
21                 if (numeros[i] > maxImp) {
22                     maxImp = numeros[i];
23                 }
24             }
25         }
26         cout<<"Arreglo: "<<endl;
27         for(int i = 0; i < 20; i++){
28             cout<<numeros[i]<<" ";
29         }
30
31         cout<<"\nSuma de los n·umeros en posiciones pares: "<<suma<<endl;
32         cout<<"N·mero m·ximo en posici·n impar: "<<maxImp<<endl;
33         cout<<"Posici·n del n·mero m·ximo par: "<<maxPar<<endl;
34
35     return 0;
36 }
```

Arreglo:

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

Suma de los n·umeros en posiciones pares: 100

N·mero m·ximo en posici·n impar: 20

Posici·n del n·mero m·ximo par: 19



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

```
[*] programa29_arreglos.cpp
1  /*Ejercicio 29*/
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     int numeros [30] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30};
8
9     int sumaPar = 0;
10    int sumaImp = 0;
11    int total = 0;
12
13    for (int i = 0; i < 30; i++) {
14        if (numeros[i] % 2 == 0) {
15            sumaPar = sumaPar + numeros[i];
16        } else {
17            sumaImp = sumaImp + numeros[i];
18        }
19        total = total + numeros[i];
20    }
21
22    cout<<"Arreglo: "<<endl;
23    for(int i = 0; i < 30; i++){
24        cout<<numeros[i]<< " ";
25    }
26
27    cout<<"\nLa suma de los números pares es: "<<sumaPar<<endl;
28    cout<<"La suma de los números impares es: "<<sumaImp<<endl;
29    cout<<"Cantidad de elementos del arreglo: "<<total<<endl;
30
31    return 0;
32 }
```

```
Arreglo:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
La suma de los números pares es: 240
La suma de los números impares es: 225
Cantidad de elementos del arreglo: 465
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

programa30\_arreglos.cpp

```
1  /*Ejercicio 30*/
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     int numeros [15] = {12, -5, 0, 8, -3, 0, 17, -9, 4, -12, 0, 6, -7, 3, 0};
8
9     int neg = 0;
10    int pos = 0;
11    int ceros = 0;
12    int sumaPos = 0;
13    int sumaNeg = 0;
14
15    for (int i = 0; i < 15; i++) {
16        if(numeros[i] < 0) {
17            neg++;
18            sumaNeg = sumaNeg + numeros[i];
19        }else if(numeros[i] > 0){
20            pos++;
21            sumaPos = sumaPos + numeros[i];
22        }else{
23            ceros++;
24        }
25    }
26    cout<<"Arreglo: "<<endl;
27    for(int i = 0; i < 15; i++){
28        cout<<numeros[i]<<" ";
29    }
30
31    cout<<"\nSuma de números negativos: "<<sumaNeg<< endl;
32    cout<<"Cantidad de números negativos: "<<neg<<endl;
33    cout<<"Suma de números positivos: "<<sumaPos<<endl;
34    cout<<"Cantidad de números positivos: "<<pos<<endl;
35    cout<<"Cantidad de ceros: "<<ceros<<endl;
```

Arreglo:

12 -5 0 8 -3 0 17 -9 4 -12 0 6 -7 3 0

Suma de números negativos: -36

Cantidad de números negativos: 5

Suma de números positivos: 50

Cantidad de números positivos: 6

Cantidad de ceros: 4



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.

```
programa31_arreglos.cpp
1  /*Arreglo 31*/
2
3 #include <iostream>
4 using namespace std;
5
6 int main(){
7     int numeros [10] = {12, -5, 0, 8, -3, 0, 17, -9, 4, -12};
8     int neg [10];
9     int pos [10];
10    int negC = 0;
11    int posC = 0;
12
13    for (int i = 0; i < 10; i++){
14        if(numeros[i]<0){
15            neg [negC] = numeros [i];
16            negC++;
17        }else{
18            pos[posC] = numeros [i];
19            posC++;
20        }
21    }
22    cout<<"Arreglo original: "<<endl;
23    for(int i = 0; i < 10; i++){
24        |cout<<numeros[i]<< " ";
25    }
26
27    cout<<"\nPositivos: "<<endl;
28    for(int i = 0; i < posC; i++){
29        |cout<<pos[i]<< " ";
30    }
31
32    cout<<"\nNegativos: "<<endl;
33    for(int i = 0; i < negC; i++){
34        |cout<<neg[i]<< " ";
35    }
```

```
Arreglo original:
12 -5 0 8 -3 0 17 -9 4 -12
Positivos:
12 0 8 0 17 4
Negativos:
-5 -3 -9 -12
```



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

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

```
programa32_arreglos.cpp
1  /*Arreglo 32*/
2
3  #include <iostream>
4  using namespace std;
5
6 int main(){
7
8     int tam;
9     cout<<"Ingrese la cantidad de elementos: "<<endl;
10    cin>>tam;
11
12    int numeros [tam];
13    int sumPar = 0;
14    int numsPar = 0;
15
16    for(int i = 0; i < tam; i++){
17        cout<<"Ingrese el número "<<i<<": "<<endl;
18        cin>>numeros[i];
19        if(i % 2 == 0){
20            numsPar++;
21            sumPar = sumPar + numeros[i];
22        }
23    }
24    double media = sumPar / numsPar;
25
26    double max = numeros [0];
27    double min = numeros [0];
28
29    for(int i = 0; i < tam; i++){
30        if(numeros[i]>max){
31            max = numeros[i];
32        }if(numeros[i]<min){
33            min = numeros[i];
34        }
35    }
36
37    int suma = 0;
38    double sumaDif = 0;
39    for(int i = 0; i < tam; i++){
40        suma = suma + numeros[i];
41        double prom = suma / tam;
42        double dif = (numeros[i] - prom) * (numeros[i] - prom);
43        sumaDif = sumaDif + dif;
44    }
45
46    double varianza = sumaDif / tam;
47
48    cout<<"\nNúmero máximo: "<<max<<endl;
49    cout<<"\nNúmero mínimo: "<<min<<endl;
50    cout<<"\nMedia de los números es posiciones pares: "<<media<<endl;
51    cout<<"\nVarianza: "<<varianza<<endl;
52
53 }
```



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.

```
programa33_arreglos.cpp
1  /*Arreglo 33*/
2
3  #include <iostream>
4  #include <cstdlib>
5  #include <ctime>
6  using namespace std;
7
8  int main()
9 {
10    int numeros [20];
11    int nums1 [20];
12    int nums2 [20];
13    int nums3 [20];
14
15    srand(time(0));
16
17    for (int i = 0; i < 20; i++) {
18        numeros [i] = rand() % 151;
19    }
20
21    for (int i = 0; i < 20 - 1; i++) {
22        for (int j = 0; j < 20 - i - 1; j++) {
23            if (numeros[j] > numeros[j + 1]) {
24                int temp = numeros[j];
25                numeros[j] = numeros[j + 1];
26                numeros[j + 1] = temp;
27            }
28        }
29    }
30
31    int nums1C = 0;
32    int nums2C = 0;
33    int nums3C = 0;
34
35    for (int i = 0; i < 20; i++) {
36
37        if (numeros[i] < 51) {
38            nums1[nums1C] = numeros[i];
39            nums1C++;
40        }else if (numeros[i] < 101){
41            nums2[nums2C] = numeros[i];
42            nums2C++;
43        }else{
44            nums3[nums3C] = numeros[i];
45            nums3C++;
46        }
47
48        cout<<"Arreglo original: "<<endl;
49        for(int i = 0; i < 20; i++){
50            cout<<numeros[i]<<" ";
51        }
52
53        cout<<"\nArreglo con n·meros del 0 al 50: "<<endl;
54        for(int i = 0; i < nums1C; i++){
55            cout<<nums1[i]<<" ";
56        }
57
58        cout<<"\nArreglo con n·meros del 51 al 100: "<<endl;
59        for(int i = 0; i < nums2C; i++){
60            cout<<nums2[i]<<" ";
61        }
62
63        cout<<"\nArreglo con n·meros mayores a 101: "<<endl;
64        for(int i = 0; i < nums3C; i++){
65            cout<<nums3[i]<<" ";
66        }
67
68    return 0;
69 }
```

```
Arreglo original:
12 16 16 18 23 29 30 41 45 45 69 94 97 108 115 118 125 128 132 148
Arreglo con n·meros del 0 al 50:
12 16 16 18 23 29 30 41 45 45
Arreglo con n·meros del 51 al 100:
69 94 97
Arreglo con n·meros mayores a 101:
108 115 118 125 128 132 148
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

```
programa34_arreglos.cpp
1  /*Arreglo 34*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main()
7 {
8     int numeros [10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
9
10    cout<<"Arreglo original: "<<endl;
11    for(int i = 0; i < 10; i++){
12        cout<<numeros[i]<< " ";
13    }
14
15    numeros[6] = numeros[0];
16
17    cout<<"\nArreglo modificado: "<<endl;
18    for(int i = 0; i < 10; i++){
19        cout<<numeros[i]<< " ";
20    }
21
22    return 0;
}
```

Arreglo original:  
1 2 3 4 5 6 7 8 9 10  
Arreglo modificado:  
1 2 3 4 5 6 1 8 9 10



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).

```
programa35_arreglos.cpp
1  /*Arreglo 35*/
2
3  #include <iostream>
4  #include <cstdlib>
5  #include <ctime>
6  #include <cmath>
7  using namespace std;
8
9  int main()
10 {
11     int numeros [20];
12
13     srand(time(0));
14
15     int suma = 0;
16     for (int i = 0; i < 20; i++) {
17         numeros [i] = rand() % 21;
18         suma = suma + numeros[i];
19     }
20     double promedio = suma/20;
21
22     for (int i = 0; i < 20 - 1; i++) {
23         for (int j = 0; j < 20 - i - 1; j++) {
24             if (numeros[j] > numeros[j + 1]) {
25                 int temp = numeros[j];
26                 numeros[j] = numeros[j + 1];
27                 numeros[j + 1] = temp;
28             }
29         }
30     }
31
32     int sumaVar = 0;
33     double sumaDif = 0;
34     for(int i = 0; i < 20; i++){
35         sumaVar = sumaVar + numeros[i];
36
37         double prom = sumaVar / 20;
38
39         double dif = (numeros[i] - prom) * (numeros[i] - prom);
40         sumaDif = sumaDif + dif;
41
42     }
43     double varianza = sumaDif / 20;
44     double desvEst = sqrt(varianza);
45
46     int masRep = numeros[0];
47     int maxRep = 1;
48     int reps = 1;
49
50     for (int i = 1; i < 20; i++) {
51         if (numeros[i] == numeros[i - 1]) {
52             reps++;
53         } else {
54             reps = 1;
55         }
56
57         if (reps > maxRep) {
58             maxRep = reps;
59             masRep = numeros[i];
60         }
61
62     }
63
64     cout<<"Arreglo: "<<endl;
65     for(int i = 0; i < 20; i++){
66         cout<<numeros[i]<<" ";
67
68     }
69
70     cout<<"\nSuma de los elementos: "<<suma<<endl;
71     cout<<"Promedio de los elementos: "<<promedio<<endl;
72     cout<<"Varianza de los elementos: "<<varianza<<endl;
73     cout<<"Desviación estándar de los elementos: "<<desvEst<<endl;
74     cout<<"Número más repetido: "<<masRep<<endl;
```

```
Arreglo:
0 0 3 4 5 6 7 7 8 10 11 14 14 15 15 16 18 18 18 19
Suma de los elementos: 208
Promedio de los elementos: 10
Varianza de los elementos: 61.3
Desviación estndar de los elementos: 7.82943
Número mss repetido: 18
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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.

programa36\_arreglos.cpp

```
1  /*Arreglo 36*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main(){
7      int numeros [10] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
8      int suma = 0;
9      for (int i = 0; i < 10; i++) {
10          suma += numeros[i];
11      }
12      double prom = suma / 10;
13
14      int nums [10];
15      int may = 0;
16      int menC = 0;
17      for (int i = 0; i < 10; i++) {
18          if (numeros[i] > prom) {
19              may++;
20          } else {
21              nums[menC] = numeros[i];
22              menC++;
23          }
24      }
25      cout<<"Arreglo original: "<<endl;
26      for(int i = 0; i < 10; i++){
27          cout<<numeros[i]<<" ";
28      }
29      cout<<"\nPromedio de todos los n meros: "<<prom<<endl;
30      cout<<"\nCantidad de n meros mayores al promedio: "<<may<<endl;
31      cout<<"\nN meros menores al promedio: "<<endl;
32      for(int i = 0; i < menC; i++){
33          cout<<nums[i]<<" ";
```

Arreglo original:

10 20 30 40 50 60 70 80 90 100

Promedio de todos los n meros: 55

Cantidad de n meros mayores al promedio: 5

N meros menores al promedio:

10 20 30 40 50



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

```
programa37_arreglos.cpp
1  /*Arreglo 37*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main()
7  {
8      int numeros [20] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20};
9
10     int cont = 0;
11     for (int i = 0; i < 20; i++){
12         int ultDig = numeros[i] % 10;
13         if(ultDig % 2 == 0){
14             cont++;
15         }
16     }
17
18     cout<<"Arreglo original: "<<endl;
19     for(int i = 0; i < 20; i++){
20         cout<<numeros[i]<< " ";
21     }
22
23     cout<<"\nCantidad de números con último dígito par: "<<cont<<endl;
24
25
26     return 0;
27 }
```

```
Arreglo original:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Cantidad de números con ·ltimo dÝgito par: 10
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

38. Convierta un número entero decimal a su equivalente en representación binaria. (

[http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2\\_contenidos\\_2c.htm](http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_contenidos_2c.htm))

```
programa38_arreglos.cpp
1  /*Arreglo 38*/
2
3  #include <iostream>
4  #include <sstream>
5  #include <string>
6  using namespace std;
7
8  int main()
9 {
10    int num;
11    cout<<"Ingrese un número: "<<endl;
12    cin>>num;
13
14    std::string binario = "";
15
16    while (num > 0){
17        int residuo = num % 2;
18        std::stringstream n;
19        n << residuo;
20        binario = n.str() + binario;
21
22        num = num / 2;
23    }
24
25    cout<<"El número en binario es: "<<binario<<endl;
26
27    return 0;
28 }
```

```
Ingrese un número:
5
El número en binario es: 101
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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

programa39\_arreglos.cpp

```
1  /*Arreglo_39*/
2
3  #include <iostream>
4  using namespace std;
5
6  int main()
7  {
8      int n;
9      cout<<"Ingrese el tamaño del arreglo: "<<endl;
10     cin>>n;
11
12     int numeros [n];
13
14    for (int i = 0; i < n; i++){
15        cout<<"Ingrese el número "<<i<<endl;
16        cin>>numeros[i];
17    }
18
19    int repeticiones[n] = {0};
20
21    for (int i = 0; i < n; i++) {
22        for (int j = 0; j < n; j++) {
23            if (numeros[i] == numeros[j]) {
24                repeticiones[i]++;
25            }
26        }
27    }
28
29    for (int i = 0; i < n; i++) {
30        bool x = false;
31        for (int j = 0; j < i; j++) {
32            if (numeros[i] == numeros[j]) {
33                x = true;
34                break;
35            }
36        }
37        if (!x) {
38            cout<<"Número: "<<numeros[i]<<endl;
39            cout<<"Cantidad de repeticiones: "<<repeticiones[i]<<endl;
40        }
41    }
42
43    return 0;
44 }
```

```
Número: 1
Cantidad de repeticiones: 2
Número: 2
Cantidad de repeticiones: 1
Número: 3
Cantidad de repeticiones: 1
Número: 4
Cantidad de repeticiones: 1
```



**UNIVERSIDAD TECNOLÓGICA  
DE ESCUINAPA**



40. Almacene en un arreglo  $n$  números ( $n$  debe ser un numero 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  $n=6, v=[1][2][3][4][5][6] v(invertido)=[3][2][1][6][5][4]$ .

programa40\_arreglos.cpp

```
1 /*Arreglo 40*/
2
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     int cant;
9     cout<<"Ingrese la cantidad de elementos para el arreglo (debe ser un número par menor a 50): "<<endl;
10    cin>>cant;
11
12    int numeros [cant];
13
14    if (cant % 2 == 0 && cant < 50){
15        for(int i = 0; i < cant; i++){
16            numeros[i] = i;
17        }
18
19        cout<<"Arreglo original: "<<endl;
20        for(int i = 0; i < cant; i++){
21            cout<<numeros[i]<<" ";
22        }
23
24        int mitad = cant / 2;
25
26        for (int i = 0; i < mitad / 2; i++) {
27            int temp = numeros[i];
28            numeros[i] = numeros[mitad - i - 1];
29            numeros[mitad - i - 1] = temp;
30        }
31
32        for (int i = mitad; i < mitad + (cant - mitad) / 2; i++) {
33            int temp = numeros[i];
34            numeros[i] = numeros[cant - 1 - (i - mitad)];
35            numeros[cant - 1 - (i - mitad)] = temp;
36        }
37
38        int resultado[cant];
39        for (int i = 0; i < cant; i++) {
40            resultado[i] = numeros[i];
41        }
42
43        cout<<"\nArreglo invertido: "<<endl;
44        for(int i = 0; i < cant; i++){
45            cout<<resultado[i]<<" ";
46        }
47
48    }else{
49        cout<<"Debes ingresar un número par menor a 50"<<endl;
50    }
51
52    return 0;
53 }
```

```
Ingrese la cantidad de elementos para el arreglo (debe ser un número par menor a 50):
20
Arreglo original:
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Arreglo invertido:
9 8 7 6 5 4 3 2 1 0 19 18 17 16 15 14 13 12 11 10
```



**UNIVERSIDAD TECNOLÓGICA  
DE ESCUINAPA**



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>

int main() {
    int a[] = {1, 6, 9, 7, 3, 4, 2, 5, 8, 10};
    int b[] = {9, 7, 6, 18, 3, 4, 8, 5, 2, 1};
    int c[10];

    for (int i = 0; i < 10; i++) {
        c[i] = a[i] + b[9 - i];
    }

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

    std::cout << "b: ";
    for (int i = 0; i < 10; i++) {
        std::cout << b[i] << " ";
    }
    std::cout << std::endl;

    std::cout << "c: ";
    for (int i = 0; i < 10; i++) {
        std::cout << c[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}
```



```
C:\Users\Cesar\Documents\R> 
a: 1 6 9 7 3 4 2 5 8 10
b: 9 7 6 18 3 4 8 5 2 1
c: 2 8 14 15 7 7 20 11 15 19
```



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>

int main() {
    int n[] = {4, 9, 8, 6, 2, 3};
    int x[] = {7, 9, 8, 3, 1, 2};
    int z[12];
    int index = 0;

    for (int i = 0; i < 6; i++) {
        z[index++] = n[i];
        z[index++] = x[i];
    }

    for (int i = 0; i < 12; i++) {
        std::cout << z[i] << " ";
    }

    return 0;
}
```

```
4 7 9 9 8 8 6 3 2 1 3 2
```



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>

int main() {
    int b[] = {1, 5, 10, 15, 4, 2, 8, 6};
    int a[] = {4, 6, 12, 18, 20, 11, 2, 4, 8, 17};
    int c[18];
    int sum = 0;

    std::cout << "b: ";
    for (int i = 7; i >= 0; i--) {
        std::cout << b[i] << " ";
    }
    std::cout << std::endl;

    for (int i = 0; i < 8; i++) {
        c[sum++] = b[i];
    }
    for (int i = 0; i < 10; i++) {
        c[sum++] = a[i];
    }

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

    std::cout << "c: ";
    for (int i = 17; i >= 0; i--) {
        std::cout << c[i] << " ";
    }
}
```



**UNIVERSIDAD TECNOLÓGICA  
DE ESCUINAPA**

b: 6 8 2 4 15 10 5 1

a: 4 6 12 18 20 11 2 4 8 17

c: 17 8 4 2 11 20 18 12 6 4 6 8 2 4 15 10 5 1



44. Imprimir las siguientes series:

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

1  
1 × 1 = 1  
11 × 11 = 121  
111 × 111 = 12321  
1111 × 1111 = 1234321  
11111 × 11111 = 123454321  
111111 × 111111 = 12345654321  
1111111 × 1111111 = 1234567654321  
11111111 × 11111111 = 123456787654321

```
#include <iostream>

int main() {
    int num = 1;
    int edit = 2;

    for (int i = 1; i < 9; i++) {
        std::cout << num << " * 9 + " << edit << " = " << (num * 9 + edit) << std::endl;
        num = num * 10 + (i + 1);
        edit++;
    }

    std::cout << std::endl;

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

return 0;
}
```



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

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>

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];
    }

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

    return 0;
}
```

**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>

int main() {
    std::string palabras[] = {"arreglo", "programa", "javat", "carro", "oso"};
    char vocales[] = {'a', 'e', 'i', 'o', 'u'};
    std::string palabrasConVocalInicial[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]) {
                palabrasConVocalInicial[count] = palabras[i];
                count++;
                break;
            }
        }
    }

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

    return 0;
}
```

Palabras que inician con una vocal: arreglo oso



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>

int main() {
    int num = 10;

    for (int i = 0; i < num; i++) {
        std::string ast = "" + i;
        for (int j = 0; j < i; j++) {
            ast += "*" + j;
        }
        std::cout << ast << std::endl;
    }

    return 0;
}
```



**UNIVERSIDAD TECNOLÓGICA  
DE ESCUINAPA**

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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;
}
```

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



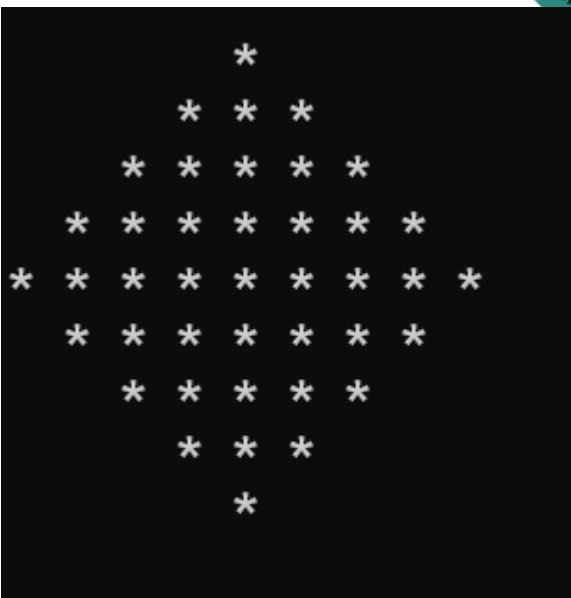
49. Escriba un programa que imprima la forma en diamante siguiente.

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

```
#include <iostream>  
  
int main() {  
    int n = 5;  
  
    for (int i = 1; i <= n; i++) {  
        for (int j = i; j < n; j++) {  
            std::cout << " ";  
        }  
        for (int j = 1; j <= (2 * i - 2); j++) {  
            std::cout << "* ";  
        }  
        std::cout << "**" << std::endl;  
    }  
  
    for (int i = n - 1; i >= 1; i--) {  
        for (int j = n; j > i; j--) {  
            std::cout << " ";  
        }  
        for (int j = 1; j <= (2 * i - 2); j++) {  
            std::cout << "* ";  
        }  
        std::cout << "**" << std::endl;  
    }  
  
    return 0;  
}
```



**UNIVERSIDAD TECNOLÓGICA  
DE ESCUINAPA**





## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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>

int main() {
    int n = 10;
    char arr[10];

    for (int i = 1; i <= n; i++) {
        for (int j = 0; j < i; j++) {
            arr[j] = '*';
        }
        for (int j = 0; j < i; j++) {
            std::cout << arr[j];
        }
        std::cout << std::endl;
    }

    std::cout << std::endl;

    for (int i = n; i >= 1; i--) {
        for (int j = 0; j < i; j++) {
            arr[j] = '*';
        }
        for (int j = 0; j < i; j++) {
            std::cout << arr[j];
        }
        std::cout << std::endl;
    }

    std::cout << std::endl;
```



```
for (int i = n; i >= 1; i--) {
    for (int j = 0; j < n - i; j++) {
        std::cout << " ";
    }
    for (int j = 0; j < i; j++) {
        arr[j] = '*';
    }
    for (int j = 0; j < i; j++) {
        std::cout << arr[j];
    }
    std::cout << std::endl;
}

std::cout << std::endl;

for (int i = 1; i <= n; i++) {
    for (int j = 0; j < n - i; j++) {
        std::cout << " ";
    }
    for (int j = 0; j < i; j++) {
        arr[j] = '*';
    }
    for (int j = 0; j < i; j++) {
        std::cout << arr[j];
    }
    std::cout << std::endl;
}

return 0;
```



# **UNIVERSIDAD TECNÓLOGICA DE ESCUINAPA**



# **UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA**



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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>
using namespace std;

int main() {
    const int x=15;
    string alum[x];
    int n[x];

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

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

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

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



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Ingrese el nombre del alumno: 0: c
Ingrese el numero favorito del alumno: c: 1
Ingrese el nombre del alumno: 1: v
Ingrese el numero favorito del alumno: v: 2
Ingrese el nombre del alumno: 2: b
Ingrese el numero favorito del alumno: b: 3
Ingrese el nombre del alumno: 3: n
Ingrese el numero favorito del alumno: n: 4
Ingrese el nombre del alumno: 4: m
Ingrese el numero favorito del alumno: m: 5
Ingrese el nombre del alumno: 5: a
Ingrese el numero favorito del alumno: a: 6
Ingrese el nombre del alumno: 6: s
Ingrese el numero favorito del alumno: s: 7
Ingrese el nombre del alumno: 7: d
Ingrese el numero favorito del alumno: d: 8
Ingrese el nombre del alumno: 8: g
Ingrese el numero favorito del alumno: g: 9
Ingrese el nombre del alumno: 9: q
Ingrese el numero favorito del alumno: q: 10
Ingrese el nombre del alumno: 10: w
Ingrese el numero favorito del alumno: w: 11
Ingrese el nombre del alumno: 11: e
Ingrese el numero favorito del alumno: e: 12
Ingrese el nombre del alumno: 12: r
Ingrese el numero favorito del alumno: r: 13
Ingrese el nombre del alumno: 13: t
Ingrese el numero favorito del alumno: t: 14
Ingrese el nombre del alumno: 14: y
Ingrese el numero favorito del alumno: y: 15
Lista de numeros favoritos ordenados de menor a mayor: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
```



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

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

int main() {
    int n = 2;
    int arr1[n];
    int arr2[n];
    int val1;
    int val2;

    for (int i = 0; i < n; i++) {
        cout << "Ingrese el numero para la posición " << i << " del arreglo 1: ";
        cin >> val1;

        if (cin.fail()) {
            cout << "Invalido\n";
            cin.clear();
            cin.ignore(1000, '\n');
            i--;
        } else {
            arr1[i] = val1;
        }
    }

    for (int i = 0; i < n; i++) {
        cout << "Ingrese el numero para la posicion " << i << " del arreglo 2: ";
        cin >> val2;

        if (cin.fail()) {
            cout << "Invalido\n";
            cin.clear();
            cin.ignore(1000, '\n');
            i--;
        } else {
            arr2[i] = val2;
        }
    }

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

    cout << "La suma total de los valores de ambos arreglos es: " << total;
    return 0;
}
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Ingrese el numero para la posicion 0 del arreglo 1: 1
Ingrese el numero para la posicion 1 del arreglo 1: 2
Ingrese el numero para la posicion 0 del arreglo 2: 3
Ingrese el numero para la posicion 1 del arreglo 2: 4
La suma total de los valores de ambos arreglos es: 10
```



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 start = 0;
    int end = 0;
    int num = 0;

    cout << "Ingrese el valor inicial: ";
    cin >> start;
    cout << "Ingrese el valor final: ";
    cin >> end;
    cout << "Ingrese el numero del que quiere saber sus múltiplos: ";
    cin >> num;

    cout << "Multiplos de " << num << " entre " << start << " y " << end << ":" ;
    for (int i = start; i <= end; i++) {
        if (i % num == 0) {
            cout << i << " ";
        }
    }
    return 0;
}
```

```
Ingrese el valor inicial: 1
Ingrese el valor final: 10
Ingrese el numero del que quiere saber sus multiplos: 2
Multiplos de 2 entre 1 y 10: 2 4 6 8 10
```



54. Pide al usuario que ingrese 5 nombres y almacenelos en un arreglo, luego muestralos 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;
}
```



Ingrese el nombre de la persona 0:  
cesar

Ingrese el nombre de la persona 1:  
gael

Ingrese el nombre de la persona 2:  
scarlet

Ingrese el nombre de la persona 3:  
tona

Ingrese el nombre de la persona 4:  
andres

Los nombres ingresados son: 0.- cesar

Los nombres ingresados son: 1.- gael

Los nombres ingresados son: 2.- scarlet

Los nombres ingresados son: 3.- tona

Los nombres ingresados son: 4.- andres



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;
}
```

```
Tamano del arreglo: 5
Ingrese el valor de la posicion 0:
1
Ingrese el valor de la posicion 1:
2
Ingrese el valor de la posicion 2:
3
Ingrese el valor de la posicion 3:
45
Ingrese el valor de la posicion 4:
5
Ingrese la posicion del valor que quiere comparar: 3
Hay 1 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;

    string nombres[n];
    cout << "Ingrese los nombres:\n";
    for (int i = 0; i < n; i++) {
        cout << "Nombre " << i << ": ";
        cin >> nombres[i];
    }
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (nombres[j] > nombres[j + 1]) {
                string temp = nombres[j];
                nombres[j] = nombres[j + 1];
                nombres[j + 1] = temp;
            }
        }
    }

    cout << "\nNombres ordenados alfabeticamente:\n";
    for (int i = 0; i < n; i++) {
        cout << nombres[i] << endl;
    }

    return 0;
}
```

```
Ingrese la cantidad de personas: 2
Ingrese los nombres:
Nombre 0: cesar
Nombre 1: moises

Nombres ordenados alfabeticamente:
cesar
moises
```



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=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<<":";
        cin>>numeros[i];
    }

    int max=numeros[0];
    for (int i=1; i<x; i++){
        if (numeros[i]>max){
            max=numeros[i];
        }
    }
    cout<<endl;
    cout<<"El valor maximo en el arreglo es: "<<max;
    return 0;
}
```

```
Tamaño del arreglo: 5
Ingrese el valor de la posicion 0:2
Ingrese el valor de la posicion 1:4
Ingrese el valor de la posicion 2:6
Ingrese el valor de la posicion 3:8
Ingrese el valor de la posicion 4:10

El valor maximo en el arreglo es: 10
```



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 <string>
using namespace std;

int main() {
    string palabra1, palabra2;

    cout << "Ingrese la primera palabra: ";
    cin >> palabra1;
    cout << "Ingrese la segunda palabra: ";
    cin >> palabra2;

    int len1 = palabra1.length();
    int len2 = palabra2.length();

    if (len1 >= 2 && len2 >= 2 && palabra1.substr(len1 - 2) == palabra2.substr(len2 - 2)) {
        cout << "Las palabras riman: " << palabra1 << " y " << palabra2 << endl;
    } else {
        cout << "Las palabras no riman" << endl;
    }

    return 0;
}
```

```
Ingrese la primera palabra: carro
Ingrese la segunda palabra: barro
Las palabras riman: carro y barro
```



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 n = 2;
    cout << "La cantidad de personas es: " << n << endl << endl;

    string noms[2];
    double hts[2];

    for (int i = 0; i < n; i++) {
        cout << "Ingrese el nombre de la persona " << i + 1 << ": ";
        cin >> noms[i];
        cout << "Ingrese la estatura de " << noms[i] << " en cm: ";
        cin >> hts[i];
        cout << endl;
    }

    cout << "Personas registradas:" << endl;
    for (int i = 0; i < n; i++) {
        cout << noms[i] << endl;
    }
    cout << endl;

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

    for (int i = 0; i < n; i++) {
        if (noms[i] == busca) {
            cout << "La estatura de " << noms[i] << " es de " << hts[i] << " cm" << endl;
            break;
        }
    }

    return 0;
}
```

```
Ingrese el nombre de la persona 1: cesar
Ingrese la estatura de cesar en cm: 172

Ingrese el nombre de la persona 2: gael
Ingrese la estatura de gael en cm: 170

Personas registradas:
cesar
gael

Ingrese el nombre de la persona para consultar su estatura: cesar
La estatura de cesar es de 172 cm
```



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 la cantidad de generos de videojuegos: ";
    cin >> n;
    if (n <= 0) {
        cout << "La cantidad de generos debe ser mayor que 0." << endl;
        return 1;
    }
    string* generos = new string[n];
    for (int i = 0; i < n; i++) {
        cout << "Ingrese el genero de videojuego " << i + 1 << ":" ;
        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 << "\nLista de generos ordenados alfabeticamente:\n";
    for (int i = 0; i < n; i++) {
        cout << generos[i] << endl;
    }

    delete[] generos;
    return 0;
}
```

```
Ingrese la cantidad de generos de videojuegos: 3
Ingrese el genero de videojuego 1: roguelike
Ingrese el genero de videojuego 2: accion
Ingrese el genero de videojuego 3: aventura

Lista de generos ordenados alfabeticamente:
accion
aventura
roguelike
```



61. Realizar un arreglo de 10 posiciones que recopile números de 1 a 10 en arábigo 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 n [10];
    string nr[] = {"I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX", "X"};

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

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

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

    return 0;
}
```

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



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

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+=ahorros[i];
    }
    cout<<endl;

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

    cout<<"\nAhorros en Dólares Americanos \n";
    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 Dólares Canadienses \n";
    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 \n";
    for(int i=0; i<12; i++){
        cout<<"Ahorros en el mes de "<<meses[i]<<": "<<ahorros[i]*yenes<<endl;
    }
```



Ingrese el ahorro para Enero: 1  
Ingrese el ahorro para Febrero: 2  
Ingrese el ahorro para Marzo: 3  
Ingrese el ahorro para Abril: 4  
Ingrese el ahorro para Mayo: 5  
Ingrese el ahorro para Junio: 6  
Ingrese el ahorro para Julio: 7  
Ingrese el ahorro para Agosto: 8  
Ingrese el ahorro para Septiembre: 9  
Ingrese el ahorro para Octubre: 10  
Ingrese el ahorro para Noviembre: 11  
Ingrese el ahorro para Diciembre: 12

**Ahorros en Pesos**

Ahorros en el mes de Enero: 1  
Ahorros en el mes de Febrero: 2  
Ahorros en el mes de Marzo: 3  
Ahorros en el mes de Abril: 4  
Ahorros en el mes de Mayo: 5  
Ahorros en el mes de Junio: 6  
Ahorros en el mes de Julio: 7  
Ahorros en el mes de Agosto: 8  
Ahorros en el mes de Septiembre: 9  
Ahorros en el mes de Octubre: 10  
Ahorros en el mes de Noviembre: 11  
Ahorros en el mes de Diciembre: 12  
Ahorro Anual: 78

**Ahorros en Euros**

Ahorros en el mes de Enero: 0.043  
Ahorros en el mes de Febrero: 0.086  
Ahorros en el mes de Marzo: 0.129  
Ahorros en el mes de Abril: 0.172  
Ahorros en el mes de Mayo: 0.215  
Ahorros en el mes de Junio: 0.258  
Ahorros en el mes de Julio: 0.301  
Ahorros en el mes de Agosto: 0.344  
Ahorros en el mes de Septiembre: 0.387  
Ahorros en el mes de Octubre: 0.43  
Ahorros en el mes de Noviembre: 0.473  
Ahorros en el mes de Diciembre: 0.516  
Ahorro Anual: 3.354



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

```
programa63_arreglos.cpp
1 #include <iostream>
2 #include <string>
3 #include <vector>
4 #include <cctype>
5 using namespace std;
6
7 int main() {
8     string palabra;
9     cout<<"Ingrese una palabra: "<<endl;
10    cin>>palabra;
11
12    for (int i = 0; i < palabra.size(); i++) {
13        palabra[i] = tolower(palabra[i]);
14    }
15
16    int v = 0;
17    int c = 0;
18    vector<char> letras(palabra.begin(), palabra.end());
19    for (int i = 0; i < letras.size(); i++) {
20        char letra = letras[i];
21        if (letra == 'a' || letra == 'e' || letra == 'i' || letra == 'o' || letra == 'u') {
22            v++;
23        } else if (isalpha(letra)) {
24            c++;
25        }
26    }
27
28    cout<<"Palabra descompuesta en un arreglo: "<<endl;
29    for (int i = 0; i < letras.size(); i++) {
30        cout<<letras[i]<< " ";
31    }
32    cout<<"\nCantidad de vocales: "<<v<<endl;
33    cout<<"Cantidad de consonantes: "<<c<<endl;
34
35    return 0;
}
```

```
Ingrese una palabra:
pato
Palabra descompuesta en un arreglo:
p a t o
Cantidad de vocales: 2
Cantidad de consonantes: 2
```

**ESTRUCTURAS DE DATOS APLICADAS**  
**UNIDAD IV – LISTAS**

**ESTUDIANTE:**  
**LÓPEZ PONCE GAELEN ANTONIO**  
**JARETH ALEXANDER LOPEZ CORONA**  
**JOSE SAUL PRADO HUIZARD**  
**4ATI**

**ESCUINAPA, SINALOA, 10 DE NOVIEMBRE DE 2024**

## LISTA 1 DINÁMICA – C++

```
1 #include <iostream>
2 #include <cstring>
3 #include <algorithm>
4 using namespace std;
5
6 struct Alumno {
7     string nombre;
8     int edad;
9     string carrera;
10 };
11
12 void Alumnos(Alumno alumnos[], int cantidad) {
13     cout << "Alumnos:\n";
14     for (int i = 0; i < cantidad; i++) {
15         cout << "Nombre: " << alumnos[i].nombre << ", Edad: " << alumnos[i].edad << ", Carrera: " << alumnos[i].carrera << endl;
16     }
17     cout << endl;
18 }
19
20 void Nombres(Alumno alumnos[], int cantidad) {
21     cout << "Nombres:\n";
22     for (int i = 0; i < cantidad; i++) {
23         cout << alumnos[i].nombre << "\n";
24     }
25     cout << "\n\n";
26 }
27
28 void ordenar(Alumno alumnos[], int cantidad) {
29     for (int i = 0; i < cantidad - 1; i++) {
30         for (int j = i + 1; j < cantidad; j++) {
31             if (alumnos[i].edad < alumnos[j].edad) {
32                 swap(alumnos[i], alumnos[j]);
33             }
34         }
35     }
36 }
37
38 void Mayores20(Alumno alumnos[], int cantidad) {
39     cout << "Alumnos mayores de 20 años:\n";
40     for (int i = 0; i < cantidad; i++) {
41         if (alumnos[i].edad > 20) {
42             cout << "Nombre: " << alumnos[i].nombre << ", Edad: " << alumnos[i].edad << ", Carrera: " << alumnos[i].carrera << endl;
43         }
44     }
45     cout << endl;
46 }
47
48 int main() {
49     int cantidad;
50     cout << "Ingrese la cantidad de alumnos: ";
51     cin >> cantidad;
52     cin.ignore();
53
54     Alumno alumnos[cantidad];
55
56     for (int i = 0; i < cantidad; i++) {
57         cout << "Ingrese el nombre del alumno " << i + 1 << ": ";
58         getline(cin, alumnos[i].nombre);
59         cout << "Ingrese la edad del alumno " << i + 1 << ": ";
60         cin >> alumnos[i].edad;
61         cin.ignore();
62         cout << "Ingrese la carrera del alumno " << i + 1 << ": ";
63         getline(cin, alumnos[i].carrera);
64     }
65
66     Alumnos(alumnos, cantidad);
67     Nombres(alumnos, cantidad);
68
69     ordenar(alumnos, cantidad);
70     cout << "Alumnos ordenados por edad:\n";
71     Alumnos(alumnos, cantidad);
72
73     Mayores20(alumnos, cantidad);
74
75     return 0;
76 }
```

```
C:\Users\gaels\OneDrive\Escr + ▾ - □ ×

Ingrese el nombre del alumno 4: victor
Ingrese la edad del alumno 4: 19
Ingrese la carrera del alumno 4: tics
Ingresar el nombre del alumno 5: ari
Ingresar la edad del alumno 5: 18
Ingresar la carrera del alumno 5: enfermeria
Alumnos:
Nombre: gael, Edad: 19, Carrera: tics
Nombre: arturo, Edad: 20, Carrera: tics
Nombre: cesar, Edad: 23, Carrera: tics
Nombre: victor, Edad: 19, Carrera: tics
Nombre: ari, Edad: 18, Carrera: enfermeria
Nombres: gael arturo cesar victor ari

Alumnos ordenados por edad:
Alumnos:
Nombre: cesar, Edad: 23, Carrera: tics
Nombre: arturo, Edad: 20, Carrera: tics
Nombre: gael, Edad: 19, Carrera: tics
Nombre: victor, Edad: 19, Carrera: tics
Nombre: ari, Edad: 18, Carrera: enfermeria

Alumnos mayores de 20 años:
Nombre: cesar, Edad: 23, Carrera: tics

-----
Process exited after 118.8 seconds with return value 0
Presione una tecla para continuar . . .
```

## LISTA 2 DINÁMICA – C++

```
1 #include <iostream>
2 #include <string>
3 #include <algorithm>
4 using namespace std;
5
6 const int NUM_CALIFICACIONES = 5;
7
8 struct Alumno {
9     string nombre;
10    string carrera;
11    double calificaciones[NUM_CALIFICACIONES];
12    int edad;
13    double promedio;
14 };
15
16 bool compararPorPromedio(const Alumno& a, const Alumno& b) {
17     return a.promedio > b.promedio;
18 }
19
20 int main() {
21     int numAlumnos;
22     cout << "Ingrese el número de alumnos: ";
23     cin >> numAlumnos;
24
25     Alumno* alumnos = new Alumno[numAlumnos];
26
27     for (int i = 0; i < numAlumnos; ++i) {
28         cout << "Alumno " << i + 1 << endl;
29         cout << "Nombre: ";
30         cin >> alumnos[i].nombre;
31         cout << "Carrera: ";
32         cin >> alumnos[i].carrera;
33         cout << "Edad: ";
34         cin >> alumnos[i].edad;
35
36         cout << "Ingrese las 5 calificaciones:" << endl;
37         double suma = 0;
38         for (int j = 0; j < NUM_CALIFICACIONES; ++j) {
39             cin >> alumnos[i].calificaciones[j];
40             suma += alumnos[i].calificaciones[j];
41         }
42         alumnos[i].promedio = suma / NUM_CALIFICACIONES;
43     }
44
45     sort(alumnos, alumnos + numAlumnos, compararPorPromedio);
46
47     cout << "\nResultados ordenados por promedio:\n";
48     for (int i = 0; i < numAlumnos; ++i) {
49         cout << "Nombre: " << alumnos[i].nombre << endl;
50         cout << "Carrera: " << alumnos[i].carrera << endl;
51         cout << "Edad: " << alumnos[i].edad << endl;
52         cout << "Promedio: " << alumnos[i].promedio << endl << endl;
53     }
54     delete[] alumnos;
55
56     return 0;
57 }
```



C:\Users\gaels\OneDrive\Escr

X

+

▼

Ingrese la edad del alumno 4: 20

Ingrese la carrera del alumno 4: tics

Ingrese el nombre del alumno 5: victor

Ingrese la edad del alumno 5: 19

Ingrese la carrera del alumno 5: tics

Alumnos:

Nombre: gael, Edad: 19, Carrera: tics

Nombre: arturo, Edad: 19, Carrera: tics

Nombre: cesar, Edad: 23, Carrera: tics

Nombre: juan, Edad: 20, Carrera: tics

Nombre: victor, Edad: 19, Carrera: tics

Nombres: gael arturo cesar juan victor

Alumnos ordenados por edad:

Alumnos:

Nombre: cesar, Edad: 23, Carrera: tics

Nombre: juan, Edad: 20, Carrera: tics

Nombre: gael, Edad: 19, Carrera: tics

Nombre: arturo, Edad: 19, Carrera: tics

Nombre: victor, Edad: 19, Carrera: tics

Alumnos mayores de 20 años:

Nombre: cesar, Edad: 23, Carrera: tics

Nombre: juan, Edad: 20, Carrera: tics