

# Iniciales:

## 1. Hello World

```
#include <iostream>

using namespace std;

int main(){
    cout << "Hello World" << endl;

    return 0;
}
```

## 2. Suma

Dame dos numeros y los sumo:

Primer numero: 3

Segundo numero: 7

El resultado es: 10

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

int suma(int num1, int num2);

int main(){
    int num1,num2;
    cout << "Dame dos numeros y los sumo: " << endl;
    cout << "Primer numero: ", cin >> num1;
    cout << "Segundo numero: ", cin >> num2;
    cout << "El resultado es: " << suma(num1, num2) << endl;
    return 0;
}

int suma(int num1, int num2){
    return num1 + num2;
}
```

## 3. Multipliación

Dame dos numeros y los multiplico:

Primer numero: 3

Segundo numero: 2

El resultado es: 6

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

int multiplicacion(int num1, int num2);

int main(){
    int num1,num2;
    cout << "Dame dos numeros y los multiplico: " << endl;
    cout << "Primer numero: ", cin >> num1;
    cout << "Segundo numero: ", cin >> num2;
    cout << "El resultado es: " << multiplicacion(num1, num2) << endl;
    return 0;
}

int multiplicacion(int num1, int num2){
    return num1 * num2;
}
```

---

## 4. Cociente y resto

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

int cociente(int num1, int num2);
int resto(int num1, int num2);

int main(){
    int num1,num2;
    cout << "Dame dos numeros y te digo el cociente y el resto: " << endl;
    cout << "Primer numero: ", cin >> num1;
    cout << "Segundo numero: ", cin >> num2;
    cout << "Su cociente es: " << cociente(num1, num2) << endl;
    cout << "Su resto es: " << resto(num1, num2) << endl;
    return 0;
}

int cociente(int num1, int num2){
    return num1 / num2;
}

int resto(int num1, int num2){
    return num1 % num2;
}
```

Dame dos numeros y te digo el cociente y el resto:  
 Primer numero: 10  
 Segundo numero: 5  
 Su cociente es: 2  
 Su resto es: 0

## 5. MCD

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

int mcd(int min_num, int max_num);

int main(){
    int num1, num2;
    cout << "Ingrese un numero: ", cin >> num1;
    cout << "Ingrese otro numero: ", cin >> num2;
    if (num1 <= num2){
        cout << "Su maximo comun divisor es: " << mcd(num1, num2);
    } else{
        cout << "Su maximo comun divisor es: " << mcd(num2, num1);
    }

    return 0;
}

int mcd(int min_num, int max_num){
    int i = min_num;
    while (i >= 1 ){
        if (min_num % i == 0 && max_num % i == 0){
            return i;
        }
        else{
            i--;
        }
    }
}
```

1 warning generated.  
 Ingrese un numero: 10  
 Ingrese otro numero: 5  
 Su maximo comun divisor es: 5%

---

## 6. Factorial

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

int factorial(int num);

int main() {
    int numero;
    cout << "Ingrese un numero y te devuelvo el factorial: ", cin >> numero;
    cout << "Su factorial es: " << factorial(numero);
}

int factorial(int num) {
    if (num == 1) {
        return 1;
    } else {
        return num * factorial(num - 1);
    }
}
```

Ingrese un numero y te devuelvo el factorial: 4  
Su factorial es: 24%

## 7. Potencia

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

float potencia(float base, int exp);

int main(){
    int num1,num2;
    cout << "Dame una base y un exponente para hacer la potencia" << endl;
    cout << "Ingrese la base: ", cin >> num1;
    cout << "Ingrese el exponente: ", cin >> num2;
    cout << "El resultado es: " << potencia(num1, num2) << endl;
    return 0;
}

float potencia(float base, int exp){
    float resultado = base;
    if (exp == 0){
        resultado = 1;
    }
    else if (exp > 0){
        while (exp > 1) {
            resultado *= base;
            exp--;
        }
    }
    else if (exp < 0){
        exp = exp * (-1);
        while (exp > 1) {
            resultado *= base;
            exp--;
        }
        resultado = 1 / resultado;
    }
    return resultado;
}
```

Dame una base y un exponente para hacer la potencia  
Ingrese la base: 2  
Ingrese el exponente: 3  
El resultado es: 8

## 8. Suma limitada (ingresar un número del 1 al 100 y se suman los dígitos del número).

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

int suma_digitos(int num);

int main(){
    int num;
    cout << "Dame un numero del 1 al 100 y te digo la suma entre los digitos" << endl;
    cout << "Ingrese el numero: ", cin >> num;
    if (num >= 1 && num <= 100){
        cout << "La suma de sus digitos es: " << suma_digitos(num);
    } else{
        cout << "Error: Numero fuera del rango establecido";
    }
    return 0;
}

int suma_digitos(int num){
    if (num == 100){
        return 1;
    }
    int cociente = num / 10;
    int resto = num % 10;
    return cociente + resto;
}
```

## 9. Longitud

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

int longitud(int num);

int main(){
    int num;
    cout << "Dame un numero y te digo su longitud: " << endl;
    cout << "Ingrese un numero: ", cin >> num;
    cout << "El resultado es: " << longitud(num) << endl;
    return 0;
}

int longitud(int num){
    int conteo = 0;
    if (num == 0){
        return 1;
    }
    while (num != 0){
        num /= 10;
        conteo++;
    }
    return conteo;
}
```

Dame un numero y te digo su longitud:  
Ingrese un numero: 1234  
El resultado es: 4

---

## 10. Suma de pares

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

int suma_pares(int num);

int main(){
    int num;
    cout << "Escribe un numero positivo y le devuelvo la suma de los digitos pares" << endl;
    cout << "Ingrese el numero: ", cin >> num;
    cout << "La suma de sus digitos es: " << suma_pares(num);
    return 0;
}

int suma_pares(int num){
    int resultado = 0;
    int i = 1;
    while (i <= num){
        if (i % 2 == 0){
            resultado += i;
        }
        i++;
    }
    return resultado;
}
```

Escribe un numero positivo y le devuelvo la suma de los digitos pares  
Ingrese el numero: 8  
La suma de sus digitos es: 20

---

## 11. Switch

```
# include "iostream"
using namespace std;

int main(){

    cout << "Ingrese la Opción a ejecutar: ";
    int opcion = 0;
    cin >> opcion;

    switch(opcion)
    {
        case 1: cout << "Usted ha seleccionado la opción 1";
        break;
        case 2: cout << "Usted ha seleccionado la opción 2";
        break;
        case 3: cout << "Usted ha seleccionado la opción 3";
        break;
        default: cout << "Usted ha ingresado una opción incorrecta";
    }

    return 0;
}
```

Ingrese la Opción a ejecutar: 2  
Usted ha seleccionado la opción 2

---

## 12. Diferencias de pasar función por valor y por referencia

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

void duplicar_por_valor(int a) {
    a = a * 2;
    cout << "Dentro de la función: " << a << endl;
}

void duplicar_por_referencia(int& a) {
    a = a * 2;
    cout << "Dentro de la función: " << a << endl;
}

int main() {
    int numero = 5;
    duplicar_por_valor(numero);
    cout << "Fuera de la función: " << numero << endl;

    duplicar_por_referencia(numero);
    cout << "Fuera de la función: " << numero << endl;

    return 0;
}
```

```
Dentro de la función: 10
Fuera de la función: 5
Dentro de la función: 10
Fuera de la función: 10
```

## Struct:

1. Crea una estructura llamada Rectángulo (campos largo y ancho). Luego, crea una función que calcule el área y el perímetro del rectángulo.

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

struct Rectangulo {
    float largo;
    float ancho;
};

float perimetro(Rectangulo valor);
float area(Rectangulo valor);

int main(){
    Rectangulo valor;
    cout << "Ingrese el valor del largo del rectangulo: ", cin >> valor.largo;
    cout << "Ingrese el valor del ancho del rectangulo: ", cin >> valor.ancho;

    cout << "El valor del perimetro es: " << perimetro(valor) << endl;
    cout << "El valor del area es: " << area(valor) << endl;
    return 0;
}

float perimetro(Rectangulo valor){
    float perimetro = valor.largo * 2 + valor.ancho * 2;
    return (perimetro);
}

float area(Rectangulo valor){
    float area = valor.largo * valor.ancho;
    return (area);
}
```

```
Ingrese el valor del largo del rectángulo: 3
Ingrese el valor del ancho del rectángulo: 2
El valor del perimetro es: 10
El valor del area es: 6
```

2. Crea una estructura llamada Tiempo (campos horas, minutos y segundos). Luego, crea una función que tome dos estructuras Tiempo y calcule la diferencia entre ambas en segundos.

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

struct Tiempo {
    int horas;
    int minutos;
    int segundos;
};

int diferencia(Tiempo tiempo1, Tiempo tiempo2);

int main(){
    Tiempo tiempo1, tiempo2;
    cout << "Ingrese la cantidad de horas del T1: ", cin >> tiempo1.horas;
    cout << "Ingrese la cantidad de minutos del T1: ", cin >> tiempo1.minutos;
    cout << "Ingrese la cantidad de segundos del T1: ", cin >> tiempo1.segundos;

    cout << "Ingrese la cantidad de horas del T2: ", cin >> tiempo2.horas;
    cout << "Ingrese la cantidad de minutos del T2: ", cin >> tiempo2.minutos;
    cout << "Ingrese la cantidad de segundos del T2: ", cin >> tiempo2.segundos;

    cout << "La diferencia de los tiempos ingresados es de " << diferencia(tiempo1, tiempo2), cout << " segundos";
    return 0;
}

int diferencia(Tiempo tiempo1, Tiempo tiempo2){
    int tiempo1EnSegundos = tiempo1.horas * 3600 + tiempo1.minutos * 60 + tiempo1.segundos;
    int tiempo2EnSegundos = tiempo2.horas * 3600 + tiempo2.minutos * 60 + tiempo2.segundos;
    if(tiempo1EnSegundos >= tiempo2EnSegundos){
        return tiempo1EnSegundos - tiempo2EnSegundos;
    } else {
        return tiempo2EnSegundos - tiempo1EnSegundos;
    }
}
```

Ingrese la cantidad de horas del T1: 3  
 Ingrese la cantidad de minutos del T1: 30  
 Ingrese la cantidad de segundos del T1: 45  
 Ingrese la cantidad de horas del T2: 6  
 Ingrese la cantidad de minutos del T2: 20  
 Ingrese la cantidad de segundos del T2: 10  
 La diferencia de los tiempos ingresados es de 10165

3. Crea una estructura llamada Fecha (campos día, mes y año). Luego, crea una función que tome dos estructuras Fecha y calcule la diferencia en días entre ambas fechas.

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

struct Fecha {
    int dia;
    int mes;
    int anno;
};

int diferencia(Fecha fecha1, Fecha fecha2);

int main(){
    Fecha fecha1, fecha2;
    cout << "Ingrese el dia de f1: ", cin >> fecha1.dia;
    cout << "Ingrese el mes de f1: ", cin >> fecha1.mes;
    cout << "Ingrese el anno de f1: ", cin >> fecha1.anno;

    cout << "Ingrese el dia de f2: ", cin >> fecha2.dia;
    cout << "Ingrese el mes de f2: ", cin >> fecha2.mes;
    cout << "Ingrese el anno de f2: ", cin >> fecha2.anno;

    cout << "La diferencia de las fechas ingresados es de aproximadamente " << diferencia(fecha1, fecha2), cout << " dias";
    return 0;
}

int diferencia(Fecha fecha1, Fecha fecha2){

    int fecha1EnDias = fecha1.anno * 365 + fecha1.mes * 30 + fecha1.dia;
    int fecha2EnDias = fecha2.anno * 365 + fecha2.mes * 30 + fecha2.dia;
    if(fecha1EnDias >= fecha2EnDias){
        return fecha1EnDias - fecha2EnDias;
    } else {
        return fecha2EnDias - fecha1EnDias;
    }
}
```

Ingrese el dia de f1: 5  
 Ingrese el mes de f1: 11  
 Ingrese el anno de f1: 2022  
 Ingrese el dia de f2: 28  
 Ingrese el mes de f2: 5  
 Ingrese el anno de f2: 2023  
 La diferencia de las fechas ingresados es de aproximadamente 208 dias

4. Crea una estructura llamada **Círculo** que tenga el campo **radio**. Luego, crea una función que calcule el área y la circunferencia del círculo.

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

struct Circulo{
    float radio;
};

float pi = 3.1416;

float area(Circulo circulo){
    return pi * circulo.radio * circulo.radio;
}

float perimetro(Circulo circulo){
    return 2 * pi * circulo.radio;
}

int main(){
    Circulo circulo;
    cout << "Ingrese el valor del radio: ", cin >> circulo.radio;
    cout << "El valor del area del circulo es " << area(circulo) << endl;
    cout << "El valor del perimetro del circulo es " << perimetro(circulo);

    return 0;
}
```

Ingrese el valor del radio: 5  
El valor del area del circulo es 78.54

5. Ingresar el nombre del alumno y su legajo

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

struct Alumno {
    string nombre;
    int legajo;
};

Alumno pedirAlUsuarioUnAlumno();

int main() {
    Alumno un_alumno = pedirAlUsuarioUnAlumno();
    cout << "El alumno fue cargado correctamente con el nombre " << un_alumno.nombre << " y su legajo es " << un_alumno.legajo;
    return 0;
}

Alumno pedirAlUsuarioUnAlumno() {
    Alumno alumno;
    cout << "Ingrese nombre del alumno: " << endl;
    cin >> alumno.nombre;
    cout << "Ingrese legajo: " << endl;
    cin >> alumno.legajo;
    return alumno;
}
```

Ingrese nombre del alumno:  
Facundo  
Ingrese legajo:  
2142851  
El alumno fue cargado correctamente con el nombre\_Facundo y su legajo es 2142851



# Array:

## Cargar frases:

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

int main() {
    // Forma 1
    char nombre[30];
    cin.getline(nombre, sizeof(nombre));

    // Forma 2
    string nombre2;
    getline(cin, nombre2);

    cout << nombre;
    cout << nombre2;
    return 0;
}
```

## Mostrar todos los elementos del Array:

```
for (int i = 0; i < size; i++){
    cout << i << ". " << array[i] << endl;
}
```

**Ejemplo:** se solicita al usuario ingresar 5 números que almacena en un array, e imprime en pantalla la suma de todos ellos.

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

int main() {
    int numeros[5];
    int suma = 0;

    for(int i = 0; i < 5; i++) {
        cout << "Ingresa el número " << i + 1 << ": ";
        cin >> numeros[i];
    }

    for(int i = 0; i < 5; i++) {
        suma += numeros[i]; // suma = suma + el número de la posición i
    }

    cout << "La suma de los números ingresados es: " << suma << endl;

    return 0;
}
```

**Ejemplo en Clase:** se solicita al usuario ingresar 5 alumnos que almacena en un array, e imprime luego en pantalla.

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

// Declaro struct y función

struct Alumno {
    string nombre;
    int edad;
};

Alumno crearAlumno();

// Código Principal
int main() {
    Alumno alumnos[5];

    for(int i = 0; i < 5; i++) {
        alumnos[i] = crearAlumno();
    }

    for(int i = 0; i < 5; i++) {
        cout << "El alumno en la posición " << i + 1 << " tiene el nombre " <<
alumnos[i].nombre << " y su edad es " << alumnos[i].edad << endl;
    }

    return 0;
}

// Definición de Función
Alumno crearAlumno() {
    Alumno alumno;
    cout << "Ingrese el nombre del alumno: " << endl;
    cin >> alumno.nombre;

    cout << "Ingrese la edad del alumno: " << endl;
    cin >> alumno.edad;

    return alumno;
}
```

---

**Ejemplo:** para obtener el tamaño de la cadena ingresada en el array. Se utiliza un ciclo **while** para recorrer el array de caracteres **cadena**, y se incrementa la variable **tamaño** en cada iteración hasta encontrar el **carácter nulo**. Luego, se muestra por pantalla el tamaño final de la cadena obtenida.

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

int main() {
    char cadena[100];
    cout << "Ingrese una cadena de caracteres: ";
    cin >> cadena;

    int tamaño = 0;
    while (cadena[tamaño] != '\0')
        tamaño++;

    cout << "La cadena ingresada tiene " << tamaño << " caracteres." << endl;
    return 0;
}
```

## Matriz:

**Ejemplo:** se solicita al usuario ingresar 9 números que almacena en una matriz 3x3, e imprime en pantalla la suma de todos ellos.

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

int main() {
    int matriz[3][3];
    int suma = 0;

    for(int i = 0; i < 3; i++) { // marca el comienzo de cada fila
        for(int j = 0; j < 3; j++) { // marca cada espacio, solo después de
                                    completarse los tres espacios de una
fila, i++

            cout << "Ingrese el elemento [" << i << "][" << j << "]: ";

            cin >> matriz[i][j];
        }

        for(int i = 0; i < 3; i++) {
            for(int j = 0; j < 3; j++){
                suma += matriz[i][j];
            }
            cout << "La suma de los elementos de la matriz es: " << suma << endl;

        return 0;
    }
}
```

**Ejemplo 2:** realizar la suma de dos matrices cuadradas de tamaño NxN.

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

const int N = 3; // Tamaño de la matriz cuadrada

int main() {
    int matriz1[N][N];
    int matriz2[N][N];
    int resultado[N][N];

    cout << "Ingrese los valores de la primera matriz: " << endl;

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++){
            cin >> matriz1[i][j];
        }
    }

    cout << "Ingrese los valores de la segunda matriz: " << endl;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++){
            cin >> matriz2[i][j];
        }
    }

    for (int i = 0; i < N; i++) { // Sumar las matrices
        for (int j = 0; j < N; j++){
            resultado[i][j] = matriz1[i][j] + matriz2[i][j];
        }
    }

    cout << "La matriz resultante es: " << endl;
    for (int i = 0; i < N; i++) { // Mostrar el resultado
        for (int j = 0; j < N; j++){
            cout << resultado[i][j];
        }
    }
    return 0;
}
```

## Otros:

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

/* Declaro funciones */
void ejecuto_accion();
void suma();
void numeros_maximo();
void triangulo();
void factorial();
int factorial(int n);
int eleccion_usuario();

/* Hago main programa */
int main() {
    int otra_accion;
    do {
        ejecuto_accion();
        cout << "Querés hacer otra cosa?" << endl;
        cout << "1. Si" << endl;
        cout << "2. No" << endl;
        cin >> otra_accion;
    } while (otra_accion == 1);
    return 0;
}

/* Defino acciones*/
int eleccion_usuario() {
    int accion_elegida;
    cout << "Qué preferís hacer?" << endl;
    cout << "1. Sumar dos números" << endl;
    cout << "2. Averiguar el tipo de triangulo segun sus lados" << endl;
    cout << "3. Averiguar el maximo de 10 números" << endl;
    cout << "4. Averiguar el factorial de un número" << endl;
    cin >> accion_elegida;
    return accion_elegida;
}

void ejecuto_accion() {
    switch (eleccion_usuario ())
    {
        case 1:
            suma ();
            break;

        case 2:
            triangulo ();
```

```

break;

case 3:
numeros_maximo ();
break;

case 4:
factorial ();
break;

default:
cout << "Opción Inválida" << endl;
}}

void suma () {
    int a;
    int b;
    cout << "Ingrese 2 números y los sumaremos" << endl;
    cout << "Ingrese el primer numero" << endl;
    cin >> a;
    cout << "Ingrese el segundo número" << endl;
    cin >> b;
    cout << "El resultado es:" << a + b << endl;
}

void numeros_maximo () {
    int maximo;
    cout << "Ingrese 10 números y le dire cuál es el máximo" << endl;
    for (int contador = 0; contador < 10; contador++) {
        int numero;
        if (contador == 0) {
            cout << "Ingrese un número:" << endl;
            cin >> numero;
            maximo = numero;
        }
        else {
            cout << "Ingrese otro número:" << endl;
            cin >> numero;
            if (numero > maximo)
                maximo = numero;
        }
    }
    cout << "El número máximo es:" << maximo << endl;
}

void triangulo () {
    int lado1;
    int lado2;
    int lado3;

```

```

cout << "Ingresa valores para lados de triangulos y te dire cuál es su tipo" << endl;
cout << "Ingresa el primer lado" << endl;
cin >> lado1;
cout << "Ingresa el segundo lado" << endl;
cin >> lado2;
cout << "Ingresa el tercer lado" << endl;
cin >> lado3;
if (lado1 == lado2 && lado2 == lado3)
    cout << "equilatero" << endl;
else if (lado1 != lado2 && lado2 != lado3 && lado1 != lado3)
    cout << "escaleno" << endl;
else
    cout << "isosceles" << endl;
}

int factorial(int n) {
    if (n > 1) {
        int resultado = n * factorial(n - 1);
        return resultado;
    }
    return 1;
}

void factorial () {
    cout << "Ingresa un número " << endl;
    int numero;
    cin >> numero;
    int resultado = factorial(numero);
    cout << "El factorial es: " << resultado << endl;
}

```

**Fechas:** hacer de fecha un entero y multiplicar por múltiplos de 10 para calcular año, mes, día.  
Restar año, mes, día cuando sea necesario.

## Búsqueda:

**Ejemplo:** Dado un array de números enteros, solicita al usuario ingresar un número y utiliza la búsqueda secuencial para determinar si ese número está presente en el array. Muestra un mensaje indicando si se encontró o no.

```

#include <iostream>
using namespace std;

bool busquedaLineal(const int array[], int size, int valorBuscado){
    for (int i = 0; i < size; i++){
        if (array[i] == valorBuscado){
            return true;           // Valor encontrado
        }
    }
    return false; // Valor no encontrado
}

```

```

}

int main(){
    int array[10] = {3, 5, 1, 6, 9, 13, 75, 98, 14, 62};
    int valorBuscado;
    cout << "Escribe un numero sofi: ", cin >> valorBuscado;
    if (busquedaLineal(array, 10, valorBuscado)){
        cout << "se encontro el nro de sofi";
    } else{
        cout << "no, segui intentando sofi";
    }
    return 0;
}

```

**Ejemplo:** Dado un array de nombres, ordenar el array y luego solicitar al usuario ingresar un nombre y utiliza la búsqueda binaria para determinar si ese nombre está presente en el array. Muestra un mensaje indicando si se encontró o no.

```

#include <iostream>
using namespace std;

void insertionSort(string nombres[], int size) {
    for (int i = 1; i < size; i++) {
        string key = nombres[i]; // Selecciona el elemento actual como el "key" para comparar e
insertar en su posición correcta
        int j = i - 1; // Inicializa un puntero al elemento anterior al actual

        // Mover los elementos mayores hacia la derecha
        while (j >= 0 && nombres[j] > key) {
            nombres[j + 1] = nombres[j]; // Mueve el elemento mayor hacia la derecha
            j--; // Decrementa el puntero para comparar con el elemento anterior
        }
        nombres[j + 1] = key; // Inserta el "key" en su posición correcta dentro del
subnombreseglo ordenado
    }}

bool busquedaBinaria(const string array[], int size, string valorBuscado) {
    int inicio = 0;
    int fin = size - 1;
    while (inicio <= fin) {
        int medio = (inicio + fin) / 2;
        if (array[medio] == valorBuscado) {
            return true; // Valor encontrado
        } else if (array[medio] < valorBuscado) {
            inicio = medio + 1; // El valor esta en la mitad derecha
        } else {
            fin = medio - 1; // El valor esta en la mitad izquierda
        }
    }
}

```



```

        return false;
    }

int main(){
    string nombres[5] = {"Roberto","Sofi","Martu","Cande","Mica"};
    insertionSort(nombres, 5);
    string nombreBuscado;
    cout << "Ingrese el nombre de la toxi: ", cin >> nombreBuscado;
    if (busquedaBinaria(nombres, 5, nombreBuscado))
    {
        cout << "La toxi ya esta ocupada";
    } else{
        cout << "La toxi esta libre";
    }
    return 0;
}

```

**Ejemplo:** Dado un array de números enteros ordenados de forma ascendente, solicita al usuario ingresar un número. Si el número ingresado es mayor que el último elemento del array, aplica la búsqueda secuencial para determinar si está presente. Si el número ingresado es menor o igual al último elemento del array, aplica la búsqueda binaria. Muestra un mensaje indicando si se encontró o no.

```

#include <iostream>
using namespace std;

bool busquedaLineal(const int array[], int size, int valorBuscado){
    for (int i = 0; i < size; i++){
        if (array[i] == valorBuscado){
            return true; // Valor encontrado
        }
    }
    return false; // Valor no encontrado
}

bool busquedaBinaria(const int array[], int size, int valorBuscado) {
    int inicio = 0;
    int fin = size - 1;
    while (inicio <= fin) {
        int medio = (inicio + fin) / 2;
        if (array[medio] == valorBuscado) {
            return true; // Valor encontrado
        } else if (array[medio] < valorBuscado) {
            inicio = medio + 1; // El valor esta en la mitad derecha
        } else {
            fin = medio - 1; // El valor esta en la mitad izquierda
        }
    }
    return false;
}

int main(){
    int array[5] = {1, 2, 3, 4, 5};

```

```

int numIngresado;
cout << "Ingrese un numero: ", cin >> numIngresado;
if (numIngresado > array[5]){
    if (busquedaLineal(array, 5, numIngresado)){
        cout << "Se encontro el numero";
    } else{
        cout << "No se encontro";
    }
} else{
    if (busquedaBinaria(array, 5, numIngresado)){
        cout << "Se encontro el numero";
    } else{
        cout << "No se encontro";
    }
}
return 0;
}

```

**Ejemplo:** a. Crea una estructura (struct) "Producto" con los atributos "nombre" y "precio".

b. Crea un array de objetos Producto y solicita al usuario ingresar un nombre de producto. Utiliza la búsqueda secuencial para determinar si ese producto está presente en el array y, en caso afirmativo, muestra su precio.

```

#include <iostream>
using namespace std;

struct Producto{
    string nombre;
    int precio;
};

int busquedaLineal(const Producto array[], int size, string valorBuscado){
    for (int i = 0; i < size; i++){
        if (array[i].nombre == valorBuscado){
            return i; // Valor encontrado
        }
    }
    return -1;
}

int main(){
    Producto producto[3];
    producto[0].nombre = "aifon";
    producto[1].nombre = "aipad";
    producto[2].nombre = "macous";
    producto[0].precio = 128332;
    producto[1].precio = 143254;
    producto[2].precio = 124563;
    string nombreBuscado;
    cout << "Ingrese un producto de apel: ", cin >> nombreBuscado;
    int i = busquedaLineal(producto, 3, nombreBuscado);
    if ( i != -1){

```

```

        cout << "El precio del " << producto[i].nombre << " es de: $"<< producto[i].precio << "
dolares.";
    } else {
        cout << "No es de apel (terrible barat)";
    }
    return 0;
}

```

**Ejemplo:** Ejercicio de búsqueda binaria en un array ordenado de fechas: Crea un array de fechas ordenadas cronológicamente. Solicita al usuario ingresar una fecha y utiliza la búsqueda binaria para determinar si esa fecha está presente en el array. Si se encuentra, muestra un mensaje indicando la posición de la fecha en el array. Para manejar fechas usar un numero con este formato: AAAAMMDD.

```

#include <iostream>
using namespace std;

int busquedaBinaria(const int array[], int size, int valorBuscado) {
    int inicio = 0;
    int fin = size - 1;
    while (inicio <= fin) {
        int medio = (inicio + fin) / 2;
        if (array[medio] == valorBuscado) {
            return medio; // Valor encontrado
        } else if (array[medio] < valorBuscado) {
            inicio = medio + 1; // El valor esta en la mitad derecha
        } else {
            fin = medio - 1; // El valor esta en la mitad izquierda
        }
    }
    return -1;
}

int main(){
    int fechaBuscada;
    int fechas[3] = {19901207, 20041123, 20110911};
    cout << "Ingresa una fecha: ", cin >> fechaBuscada;
    int i = busquedaBinaria(fechas, 3, fechaBuscada);
    if (i != -1){
        cout << "La fecha ingresada esta en la posicion nro: " << i + 1;
    } else{
        cout << "La fecha ingresada no esta en el array";
    }
}

```

**Ejemplo:** Crea una estructura llamada "Producto" con los atributos "nombre", "precio" y "stock" e implementa un sistema donde el usuario pueda ingresar los datos de 10 productos. Luego, el sistema solicitará al usuario que ingrese el nombre de un producto a buscar. Utilizando un algoritmo de búsqueda adecuado, se determinará si el producto está presente en el conjunto de productos cargados. Si se encuentra, se mostrará su nombre, precio y stock. Además, se brindará información adicional según el enunciado.

```

#include <iostream>
using namespace std;

struct Producto{

```

```

    string nombre;
    int precio;
    int stock;
};

Producto crearProducto() {
    Producto producto;
    cout << "Ingrese el nombre del producto: " << endl;
    cin >> producto.nombre;
    cout << "Ingrese el precio del producto: " << endl;
    cin >> producto.precio;
    cout << "Ingrese el stock del producto: " << endl;
    cin >> producto.stock;
    return producto;
}

int busquedaLineal(const Producto array[], int size, string valorBuscado) {
    for (int i = 0; i < size; i++) {
        if (array[i].nombre == valorBuscado) {
            return i; // Valor encontrado
        }
    }
    return -1;
}

int main() {
    Producto producto[10];
    for (int i = 0; i < 10 ; i++) {
        producto[i] = crearProducto();
    }
    string productoBuscado;
    cout << "Ingrese el nombre del producto: ", cin >> productoBuscado;
    int i = busquedaLineal(producto, 10, productoBuscado);
    if (i != -1) {
        cout << "El precio del " << producto[i].nombre << " es de $" << producto[i].precio << "
dolares y contamos con " << producto[i].stock << " en stock.";
    } else {
        cout << "No es de apel (terrible barat)";
    }
    return 0;
}

```

**Ejemplo:** Crea una estructura llamada "Instagamer" con los atributos "nombre", "edad" y una estructura anidada llamada "Redes" con los atributos "instagram" y "twitter" e implementa un sistema donde el usuario pueda ingresar los datos de 10 Instagamers. Luego, el sistema solicitará al usuario que ingrese el nombre de un Instagamer a buscar. Utilizando un algoritmo de búsqueda adecuado, se determinará si el Instagamer está presente en el conjunto de Instagamers cargados. Si se encuentra, se mostrará su nombre, edad, instagram y twitter. Además, se brindará información adicional según el enunciado.

```

#include <iostream>

using namespace std;

```

```

struct Redes{
    string instagram;
    string twitter;
};

struct Instagamer{
    string nombre;
    int edad;
    Redes redes;
};

Instagamer crearInstagamer() {
    Instagamer instagamer;
    cout << "Ingresa el nombre del instagamer: ", cin >> instagamer.nombre;
    cout << "Ingresa la edad del instagamer: ", cin >> instagamer.edad;
    cout << "Ingresa el instagram del instagamer: ", cin >> instagamer.redes.instagram;
    cout << "Ingresa el twitter del instagamer: ", cin >> instagamer.redes.twitter;
    return instagamer;
}

int busquedaLineal(const Instagamer array[], int size, string valorBuscado){
    for (int i = 0; i < size; i++) {
        if (array[i].nombre == valorBuscado){
            return i; // Valor encontrado
        }
    }
    return -1;
}

int main(){
    Instagamer instagamer[10];
    for (int i = 0; i < 10 ; i++){
        instagamer[i] = crearInstagamer();
    }
    string instagamerBuscado;
    cout << "Ingresa el nombre del instagamer que desea buscar: ", cin >> instagamerBuscado;
    int i = busquedaLineal(instagamer, 10, instagamerBuscado);
    if (i != -1){
        cout << "La edad del instagamer " << instagamer[i].nombre << " es de " <<
instagamer[i].edad << " años. "
        << "Su instagram es " << instagamer[i].redes.instagram
        << " y su twitter " << instagamer[i].redes.twitter << ".";
    } else {
        cout << "El instagamer se encuentra con la toxi";
    }
    return 0;
}

```

**Ejemplo:** Crea una estructura llamada "Factura" con los atributos "numero", "fecha" y "monto" e implementa un sistema donde el usuario pueda ingresar los datos de 10 facturas. Luego, el sistema solicitará al usuario que ingrese un número de factura a buscar. Utilizando un algoritmo de búsqueda adecuado, se determinará si la factura está presente en el conjunto de facturas cargadas. Si se encuentra, se mostrará su número, fecha y monto. Además, se brindará información adicional según el enunciado.

```
#include <iostream>

using namespace std;

struct Factura{
    int numero;
    int fecha;
    int monto;
};

Factura crearFactura() {
    Factura factura;
    cout << "Ingrese el numero de la factura: "<< endl;
    cin >> factura.numero;
    cout << "Ingrese el monto de la factura: "<< endl;
    cin >> factura.monto;
    cout << "Ingrese la fecha de la factura: "<< endl;
    cin >> factura.fecha;
    return factura;
}

int busquedaLineal(const Factura array[], int size, int facturaBuscada){
    for (int i = 0; i < size; i++) {
        if (array[i].numero == facturaBuscada) {
            return i; // Valor encontrado
        }
    }
    return -1;
}

int main(){
    Factura factura[10];
    for (int i = 0; i < 10 ; i++){
        factura[i] = crearFactura();
    }
    int facturaBuscada;
    cout << "Ingrese el numero del factura a buscar: ", cin >> facturaBuscada;
    int i = busquedaLineal(factura, 10, facturaBuscada);
    if (i != -1){
        cout << "La fecha de la factura nro " << factura[i].numero << " es del "<<
factura[i].fecha << " y posee un monto de $" << factura[i].monto << ".";
    } else {
        cout << "Factura no encontrada.";
    }
    return 0;
}
```

```
}
```

**Ejemplo:** Dada una estructura llamada Alumno con los atributos de 'nombre', 'edad' y 'promedio', implementa un sistema donde el usuario pueda ingresar los datos de 10 alumnos de una clase. Luego, el sistema solicitará al usuario que ingrese el nombre de un alumno a buscar. Utilizando la búsqueda que considere correspondiente, se determinará si el alumno está presente en el conjunto de alumnos cargados. Si se encuentra, se mostrará su nombre, edad y promedio. Además, se informará la cantidad de alumnos que tienen un promedio mayor que el del alumno buscado y la cantidad de alumnos que tienen un promedio menor que el del alumno buscado.

```
#include <iostream>
```

```
using namespace std;
```

```
struct Alumno{
    string nombre;
    int edad;
    int promedio;
};
```

```
Alumno crearAlumno() {
    Alumno alumno;
    cout << "Ingresa el nombre del alumno: ", cin >> alumno.nombre;
    cout << "Ingresa la edad del alumno: ", cin >> alumno.edad;
    cout << "Ingresa el promedio del alumno: ", cin >> alumno.promedio;
    return alumno;
}
```

```
// Función para buscar un alumno en el arreglo y retornar su índice
```

```
int busquedaAlumno(Alumno array[], int size, string alumnoBuscado) {
    for (int i = 0; i < size; i++) {
        if (array[i].nombre == alumnoBuscado){
            return i; // Valor encontrado
        }
    }
    return -1; // Valor no encontrado
}
```

```
// Función para buscar la cantidad de alumnos con promedio superior a un valor dado
```

```
int busquedaPromedioSuperior(Alumno array[], int size, int promedio) {
    int resultado = 0;
    for (int i = 0; i < size; i++){
        if (array[i].promedio > promedio){
            resultado++;
        }
    }
    return resultado;
}
```

```
// Función para buscar la cantidad de alumnos con promedio inferior a un valor dado
```

```
int busquedaPromedioInferior(Alumno array[], int size, int promedio) {
    int resultado = 0;
    for (int i = 0; i < size; i++){
        if (array[i].promedio < promedio){
```

```

        resultado++;
    } }
    return resultado;
}

int main() {

    Alumno alumno[10];
    // Solicitar datos de 10 alumnos
    for (int i = 0; i < 10 ; i++){
        alumno[i] = crearAlumno();
    }

    string alumnoBuscado;
    cout << "Ingrese el nombre del alumno a buscar: ", cin >> alumnoBuscado;
    // Buscar el alumno ingresado
    int i = busquedaAlumno(alumno, 10, alumnoBuscado);
    if (i != -1){
        cout << "El alumno " << alumno[i].nombre << " tiene:" << endl;
        cout << alumno[i].edad << " annos." << endl;
        cout << alumno[i].promedio << " de promedio." << endl;
        cout << "Hay " << busquedaPromedioSuperior(alumno, 3, alumno[i].promedio) << " alumnos
con mayor promedio, y "
        << busquedaPromedioInferior(alumno, 3, alumno[i].promedio) << " alumnos con menor
promedio.";
    } else {
        cout << "Alumno no encontrado.";
    }
    return 0;
}

```

**Ejemplo:** Crea una estructura llamada "Cliente" con los atributos "nombre", "edad" y "saldo" e implementa un sistema donde el usuario pueda ingresar los datos de 10 clientes. Luego, el sistema solicitará al usuario que ingrese el nombre de un cliente a buscar. Utilizando un algoritmo de búsqueda adecuado, se determinará si el cliente está presente en el conjunto de clientes cargados. Si se encuentra, se mostrará su nombre, edad y saldo y cuantos clientes tienen un saldo más alto que el cliente encontrado.

```

#include <iostream>
using namespace std;

struct Cliente{
    string nombre;
    int edad;
    int saldo;
};

Cliente crearCliente() {
    Cliente cliente;
    cout << "Ingrese el nombre del cliente: " << endl;
    cin >> cliente.nombre;
}

```



```

cout << "Ingrese la edad del cliente: "<< endl;
cin >> cliente.edad;

cout << "Ingrese el saldo del cliente: "<< endl;
cin >> cliente.saldo;
return cliente;
}

// Función para buscar un cliente en el arreglo y retornar su índice
int busquedacliente(Cliente array[], int size, string clienteBuscado) {
    for (int i = 0; i < size; i++) {
        if (array[i].nombre == clienteBuscado){
            return i; // Valor encontrado
        }
    }
    return -1; // Valor no encontrado
}

// Función para buscar la cantidad de clientes con saldo superior a un valor dado
int busquedaSaldo(Cliente array[], int size, int saldo) {
    int resultado = 0;
    for (int i = 0; i < size; i++){
        if (array[i].saldo > saldo){
            resultado++;
        }
    }
    return resultado;
}

int main() {
    Cliente cliente[10];
    for (int i = 0; i < 10 ; i++){
        cliente[i] = crearCliente();
    }

    string clienteBuscado;
    cout << "Ingrese el nombre del cliente a buscar: ", cin >> clienteBuscado;
    // Buscar el cliente ingresado
    int i = busquedacliente(cliente, 10, clienteBuscado);
    if (i != -1){
        cout << "El cliente " << cliente[i].nombre << " tiene:" << endl;
        cout << cliente[i].edad << " annos." << endl;
        cout << "$" << cliente[i].saldo << " de saldo." << endl;
        cout << "Hay " << busquedaSaldo(cliente, 3, cliente[i].saldo) << " clientes con mayor
saldo.";
    } else {
        cout << "Cliente no encontrado.";
    }
    return 0;
}

```

## Chararray:

**Ejemplo:** Se requiere un programa que permita al usuario ingresar una frase y mostrar por pantalla cada palabra en una línea separada.

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

int main(){
    string frase;
    cout << "Ingrese una frase: ";
    getline(cin, frase);
    int length = sizeof(frase) / sizeof(frase[0]);
    for (int i = 0; i < length; i++){
        if (frase[i] != ' ' && frase[i] != '\0')
            cout << frase[i];
        else if (frase[i] == ' ' && frase[i] != '\0')
            cout << endl;
        else break;
    }
    return 0;
}
```

**Ejemplo:** Se requiere un programa que permita al usuario ingresar una frase y mostrar por pantalla la frase invertida.

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

int main() {
    char palabra[256]; // Declaración de un arreglo de caracteres llamado 'palabra' con
    capacidad para 256 caracteres
    cout << "Ingrese una palabra: ";
    cin >> palabra;
    cout << "La palabra invertida es: ";
    int tamaño = 0;
    while (palabra[tamaño] != '\0') {
        tamaño++; // Identifica el tamaño de la palabra
    }
    for (int i = tamaño; i >= 0; i--) {
        cout << palabra[i]; // Imprimir el carácter actual en la posición 'i' del arreglo
    }
}
```

---

**Ejemplo:** Tenemos un cliente que quiere realizar un simulador de desempeño de ventas en supermercados. El usuario debe ingresar información de 2 supermercados, cada uno con 5 vendedores. Cada vendedor tiene un nombre y una cantidad de ventas realizadas. Una vez cargados ambos supermercados, el sistema muestra por pantalla el resultado de su desempeño.

¿Cómo calcula el resultado? Bajo diferentes criterios:

- El supermercado que tiene la mayor cantidad de ventas suma 2 puntos en su puntaje.
- El supermercado que tiene al mejor vendedor suma 1 punto en su puntaje.
- El supermercado que tiene al peor vendedor resta 1 punto en su puntaje (solo si tiene puntos a favor).
- Comparando vendedor por vendedor según su desempeño, se suma un punto por cada vendedor mejor. Es decir, se compara al mejor vendedor de un supermercado contra el mejor vendedor del otro y suma un punto el de mayor cantidad de ventas. Luego, se hace lo mismo con el segundo mejor vendedor de cada supermercado y suma un punto el de mayor desempeño. Luego, se repite el proceso con el tercer, cuarto y quinto mejor vendedor de ambos supermercados.

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

struct Vendedor {
    string nombre;
    int ventas;
};

void pedirleAlUsuarioQueCargueSupermercado(Vendedor[],int);
Vendedor pedirleAlUsuarioQueMeDeUnVendedor(int);
void insertarVendedorOrdenado(Vendedor[], int, Vendedor);
void mostrarResultados(int, int);
int damePuntosMasVentas(Vendedor[], Vendedor [],int);
int damePuntosMejorVendedor(Vendedor[],Vendedor[]);
int damePuntosPeorVendedor(Vendedor[],Vendedor [],int);
int damePuntosComparacionVendedores(Vendedor[],Vendedor [],int);
int calcularPuntosSupermercado(Vendedor[], Vendedor[], int);

int main() {
    const int CANTIDAD_VENEDORES = 5;
    Vendedor supermercado1[CANTIDAD_VENEDORES];
    Vendedor supermercado2[CANTIDAD_VENEDORES];
    cout << "INGRESE LOS DATOS DEL PRIMER SUPERMERCADO: " << endl;
    pedirleAlUsuarioQueCargueSupermercado(supermercado1,CANTIDAD_VENEDORES);
    cout << "INGRESE LOS DATOS DEL SEGUNDO SUPERMERCADO: " << endl;
    pedirleAlUsuarioQueCargueSupermercado(supermercado2,CANTIDAD_VENEDORES);

    int puntosSupermercado1 = calcularPuntosSupermercado(supermercado1, supermercado2,
CANTIDAD_VENEDORES);
    int puntosSupermercado2 = calcularPuntosSupermercado(supermercado2, supermercado1,
CANTIDAD_VENEDORES);

    mostrarResultados(puntosSupermercado1, puntosSupermercado2);
    return 1;
}

void mostrarResultados(int puntosSupermercado1, int puntosSupermercado2) {
    cout << "Puntos supermercado 1: " << puntosSupermercado1 << endl;
    cout << "Puntos supermercado 2: " << puntosSupermercado2 << endl;
}
```

```

void insertarVendedorOrdenado(Vendedor arr[], int size, Vendedor element) {
    int i = size - 1;
    while (i >= 0 && arr[i].ventas < element.ventas) {
        arr[i + 1] = arr[i];
        i--;
    }
    arr[i + 1] = element;
}

Vendedor pedirleAlUsuarioQueMeDeUnVendedor(int i) {
    Vendedor vendedor;
    cout << "Ingrese el nombre del vendedor nro " << i + 1 << ": " << endl;
    cin >> vendedor.nombre;
    cout << "Ingrese las ventas del vendedor nro " << i + 1 << ": " << endl;
    cin >> vendedor.ventas;
    return vendedor;
}

void pedirleAlUsuarioQueCargueSupermercado(Vendedor supermercado[], int cantidad) {
    for (int i = 0; i < cantidad; i++) {
        Vendedor vendedor = pedirleAlUsuarioQueMeDeUnVendedor(i);
        insertarVendedorOrdenado(supermercado, i, vendedor);
    }
}

int damePuntosMasVentas(Vendedor supermercado1[], Vendedor supermercado2[], int
cantidadVendedores) {
    int ventasSupermercado1 = 0;
    int ventasSupermercado2 = 0;
    int puntos = 0;
    for (int i = 0; i < cantidadVendedores; i++) {
        ventasSupermercado1 = ventasSupermercado1 + supermercado1[i].ventas;
        ventasSupermercado2 = ventasSupermercado2 + supermercado2[i].ventas;
    }
    if (ventasSupermercado1 > ventasSupermercado2)
        puntos = 2;
    return puntos;
}

int damePuntosMejorVendedor(Vendedor supermercado1[], Vendedor supermercado2[]) {
    int puntos = 0;

    if (supermercado1[0].ventas > supermercado2[0].ventas)
        puntos = 1;
    return puntos;
}

int damePuntosComparacionVendedores(Vendedor supermercado1[], Vendedor supermercado2[], int
cantidadVendedores) {

```

```

int puntos = 0;
for (int i = 0; i < cantidadVendedores; i++) {
    if (supermercado1[i].ventas > supermercado2[i].ventas)
        puntos++;
}
return puntos;
}

int damePuntosPeorVendedor(Vendedor supermercado1[], Vendedor supermercado2[], int
cantidadVendedores) {
    int puntos = 0;
    if (supermercado1[cantidadVendedores - 1].ventas < supermercado2[cantidadVendedores -
1].ventas)
        puntos = 1;
    return puntos;
}

int calcularPuntosSupermercado(Vendedor supermercado1[], Vendedor supermercado2[], int
cantidadVendedores) {
    int puntosMasventas = damePuntosMasVentas(supermercado1, supermercado2, cantidadVendedores);
    int puntosMejorVendedor = damePuntosMejorVendedor(supermercado1, supermercado2);
    int puntosPorComparacionVendedores =
damePuntosComparacionVendedores(supermercado1, supermercado2, cantidadVendedores);
    int puntos = puntosMasventas + puntosMejorVendedor + puntosPorComparacionVendedores;
    if (puntos > 0)
        puntos = puntos -
damePuntosPeorVendedor(supermercado1, supermercado2, cantidadVendedores);
    return puntos;
}

```

## Fulbo: agus:

- El equipo que suma más calidad en sus jugadores suma 2 goles.
- El equipo que tiene al mejor jugador suma 1 gol
- Comparando jugador por jugador según su nivel, se suma un gol por cada jugador mejor.

Es decir, se compara al mejor de un equipo contra el mejor del otro y suma gol el de mayor calidad.

Luego, se hace lo mismo con el segundo de cada equipo y suma gol el de mayor calidad.

Luego, se hace lo mismo con el tercero, cuarto y quinto.

- El equipo que tiene al peor jugador resta 1 gol (solo si tiene goles a favor)

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

struct Jugador{
    string nombre;
    int calidad;
};

void insertarOrdenado(Jugador arr[], int size, Jugador element) {
    int i = size - 1;
    // Desplazar elementos menores hacia la derecha
    while (i >= 0 && arr[i].calidad < element.calidad) {
        arr[i + 1] = arr[i]; // Mueve el elemento actual hacia la derecha
        i--; // Decrementa el índice para comparar con el elemento anterior
    }
    // Insertar el elemento en su posición correcta
    arr[i + 1] = element;
}

Jugador crearJugador(int i){
    Jugador jugador;
    cout << "Ingrese el nombre del jugador nro" << i + 1 << ": ", cin >> jugador.nombre;
    cout << "Ingrese la calidad del jugador nro" << i + 1 << ": ", cin >> jugador.calidad;
    return jugador;
}

void crearEquipo(Jugador array[], int size){
    for(int i=0; i < size; i++){
        Jugador jugador = crearJugador(i);
        insertarOrdenado(array, i, jugador);
    }
}

void mostrarArray(Jugador array[], int size) {
    for (int i = 0; i < size; i++) {
        cout << i << ". " << array[i].nombre << endl;
        cout << i << ". " << array[i].calidad << endl;
    }
}

void golesCalidad(Jugador equipo1[], Jugador equipo2[], int size, int& golesEquipo1, int& golesEquipo2) {
```

```

int calidadEquipol = 0, calidadEquipo2 = 0;
for(int i = 0; i < size; i++){
    calidadEquipol += equipol[i].calidad;
    calidadEquipo2 += equipo2[i].calidad;
}
if (calidadEquipol > calidadEquipo2){
    golesEquipol += 2;
} else if (calidadEquipol < calidadEquipo2){
    golesEquipo2 += 2;
}

void golesMejorJugador(Jugador equipol[], Jugador equipo2[], int& golesEquipol, int&
golesEquipo2){
    if(equipol[0].calidad > equipo2[0].calidad){
        golesEquipol += 1;
    } else if(equipol[0].calidad < equipo2[0].calidad){
        golesEquipo2 += 1;
    }
}

void golesComparacion(Jugador equipol[], Jugador equipo2[], int size, int& golesEquipol, int&
golesEquipo2){
    for(int i = 0; i < size; i++){
        if (equipol[i].calidad > equipo2[i].calidad){
            golesEquipol += 1;
        } else if (equipol[i].calidad < equipo2[i].calidad){
            golesEquipo2 += 1;
        }
    }
}

void golesPeorJugador(Jugador equipol[], Jugador equipo2[], int size, int& golesEquipol, int&
golesEquipo2){
    if(equipol[size-1].calidad > equipo2[size-1].calidad && golesEquipo2 > 0){
        golesEquipo2 -= 1;
    } else if(equipol[size-1].calidad < equipo2[size-1].calidad && golesEquipol > 0){
        golesEquipol -= 1;
    }
}

void mostrarResultado(int golesEquipol, int golesEquipo2){
    cout << "El resultado del partido sera: Equipol " << golesEquipol << " - " << golesEquipo2
<< " Equipo2";
}

void calcularGoles(Jugador equipol[], Jugador equipo2[], int size, int& golesEquipol, int&
golesEquipo2){
    golesCalidad(equipol, equipo2, size, golesEquipol, golesEquipo2);
    golesMejorJugador(equipol, equipo2, golesEquipol, golesEquipo2);
    golesComparacion(equipol, equipo2, size, golesEquipol, golesEquipo2);
    golesPeorJugador(equipol, equipo2, size, golesEquipol, golesEquipo2);
}

```

```
int main(){
    const int CANTIDAD_JUGADORES = 5;
    Jugador equipo1[CANTIDAD_JUGADORES];
    Jugador equipo2[CANTIDAD_JUGADORES];
    cout << "Ingrese los jugadores del primer equipo: " << endl;
    crearEquipo(equipo1, CANTIDAD_JUGADORES);
    cout << "Ingrese los jugadores del segundo equipo: " << endl;
    crearEquipo(equipo2, CANTIDAD_JUGADORES);
    int golesEquipo1 = 0, golesEquipo2 = 0;
    calcularGoles(equipo1, equipo2, CANTIDAD_JUGADORES, golesEquipo1, golesEquipo2);
    mostrarResultado(golesEquipo1, golesEquipo2);
    return 0;
}
```

## profe:

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

struct Jugador {
    string nombre;
    int calidad;
};

void pedirleAlUsuarioQueCargueEquipo(Jugador[],int);
void cargarEquipoFake(Jugador[],int,int);
int calcularGolesEquipo(Jugador[], Jugador[], int);
void mostrarResultados(int, int);
void mostrarEquipo(Jugador[],int);

int main() {
    const int CANTIDAD_JUGADORES = 2;
    Jugador equipo1[CANTIDAD_JUGADORES];
    Jugador equipo2[CANTIDAD_JUGADORES];

    pedirleAlUsuarioQueCargueEquipo(equipo1,CANTIDAD_JUGADORES); //Asegurarse que salga ordenado
    pedirleAlUsuarioQueCargueEquipo(equipo2,CANTIDAD_JUGADORES);
    int golesEquipo1 = calcularGolesEquipo(equipo1, equipo2, CANTIDAD_JUGADORES);
    int golesEquipo2 = calcularGolesEquipo(equipo2, equipo1, CANTIDAD_JUGADORES);
    mostrarResultados(golesEquipo1, golesEquipo2);
    return 1;
}

void mostrarResultados(int golesEquipo1, int golesEquipo2) {
    cout << "Goles equipo 1: " << golesEquipo1 << endl;
    cout << "Goles equipo 2: " << golesEquipo2 << endl;
}
```



```

void insertarJugadorOrdenado(Jugador arr[], int size, Jugador element) {
    int i = size - 1;
    // Desplazar elementos mayores hacia la derecha
    while (i >= 0 && arr[i].calidad < element.calidad) {
        arr[i + 1] = arr[i];
        i--;
    }
    // Insertar el elemento en su posición correcta
    arr[i + 1] = element;
}

Jugador pedirleAlUsuarioQueMeDeUnJugador() {
    Jugador jugador;
    cout << "Ingrese el nombre del jugador (solo uno)" << endl;
    cin >> jugador.nombre;
    cout << "Ingrese la calidad del jugador" << endl;
    cin >> jugador.calidad;
    return jugador;
}

void pedirleAlUsuarioQueCargueEquipo(Jugador equipo[], int cantidad) {
    for (int i = 0; i < cantidad; i++) {
        Jugador jugador = pedirleAlUsuarioQueMeDeUnJugador();
        insertarJugadorOrdenado(equipo, i, jugador);
    }
}

int dameGolesMasCalidad(Jugador equipo1[], Jugador equipo2[], int cantidadJugadores) {
    int calidadEquipo1 = 0;
    int calidadEquipo2 = 0;
    int goles = 0;
    for (int i = 0; i < cantidadJugadores; i++) {
        calidadEquipo1 = calidadEquipo1 + equipo1[i].calidad;
        calidadEquipo2 = calidadEquipo2 + equipo2[i].calidad;
    }

    if (calidadEquipo1 > calidadEquipo2)
        goles = 2;
    return goles;
}

int dameGolesMejorJugador(Jugador equipo1[], Jugador equipo2[]) {
    int goles = 0;
    if (equipo1[0].calidad > equipo2[0].calidad)
        goles = 1;
    return goles;
}

```

```

int dameGolesComparacionJugadores(Jugador equipo1[], Jugador equipo2[], int cantidadJugadores) {
    int goles = 0;
    for (int i = 0; i < cantidadJugadores; i++) {
        if (equipo1[i].calidad > equipo2[i].calidad)
            goles++;
    }
    return goles;
}

int golesPeorJugador(Jugador equipo1[], Jugador equipo2[], int cantidadJugadores) {
    int goles = 0;
    if (equipo1[cantidadJugadores - 1].calidad < equipo2[cantidadJugadores - 1].calidad)
        goles = 1;
    return goles;
}

int calcularGolesEquipo(Jugador equipo1[], Jugador equipo2[], int cantidadJugadores) {
    int golesMasCalidad = dameGolesMasCalidad(equipo1, equipo2, cantidadJugadores);
    int golesMejorJugador = dameGolesMejorJugador(equipo1, equipo2);
    int golesPorComparacionJugadores =
dameGolesComparacionJugadores(equipo1, equipo2, cantidadJugadores);
    int goles = golesMasCalidad + golesMejorJugador + golesPorComparacionJugadores;
    if (goles > 0)
        goles = goles - golesPeorJugador(equipo1, equipo2, cantidadJugadores);
    return goles;
}

```

**fulbo2:** Al cliente el gusto mucho nuestro sistema, y quiere hacer cambios para llevarlo a una liga profesional. Los cambios que requiere son:

- Son equipos de 11 Jugadores
- Que equipo tenga un nombre y una cantidad de socios.
- Que el equipo con más socios sume 1 gol
- Que cuando muestra el resultado lo diga con el siguiente formato.
  - Gano el equipo NOMBRE GOLES\_GANADOR - GOLES\_PERDEDOR
  - Empataron los equipos NOMBRE1 y NOMBRE2 GOLES\_EMPATE - GOLES\_EMPATE

```

#include <iostream>
using namespace std;

```

```

struct Jugador{
    string nombre;
    int calidad;
};

```

```

struct Equipo {
    string nombre_equipo;
    Jugador jugadores[2];
    int socios;
}

```

```
};

Jugador crearJugador(int);
Equipo crearEquipo(int);
void insertarOrdenado(Jugador[], int, Jugador);
void golesCalidad(Equipo, Equipo, int, int&, int&);
void golesMejorJugador(Equipo, Equipo, int&, int&);
void golesPeorJugador(Equipo, Equipo, int, int&, int&);
void golesComparacion(Equipo, Equipo, int, int&, int&);
void calcularGoles(Equipo, Equipo, int, int&, int&);
void mostrarResultado(Equipo, Equipo, int, int);
void golesPorSocio(Equipo, Equipo, int&, int&);

int main(){
    const int CANTIDAD_JUGADORES = 2;
    cout << "INGRESE LOS DATOS DEL PRIMER EQUIPO: " << endl;
    Equipo equipo1 = crearEquipo(CANTIDAD_JUGADORES);
    cout << "INGRESE LOS DATOS DEL SEGUNDO EQUIPO: " << endl;
    Equipo equipo2 = crearEquipo(CANTIDAD_JUGADORES);
    int golesEquipo1 = 0, golesEquipo2 = 0;
    calcularGoles(equipo1, equipo2, CANTIDAD_JUGADORES, golesEquipo1, golesEquipo2);
    mostrarResultado(equipo1, equipo2, golesEquipo1, golesEquipo2);
    return 0;}

void insertarOrdenado(Jugador arr[], int size, Jugador element) {
    int i = size - 1;
    // Desplazar elementos menores hacia la derecha
    while (i >= 0 && arr[i].calidad < element.calidad) {
        arr[i + 1] = arr[i]; // Mueve el elemento actual hacia la derecha
        i--; // Decrementa el índice para comparar con el elemento anterior
    }
    // Insertar el elemento en su posición correcta
    arr[i + 1] = element;
}

Equipo crearEquipo(int size){
    Equipo equipo;
    cout << "Ingrese nombre del equipo:" << endl;
    cin >> equipo.nombre_equipo;
    cout << "Ingrese numero de socios:" << endl;
    cin >> equipo.socios;
    for(int i=0; i < size; i++){
        Jugador jugador = crearJugador(i);
        insertarOrdenado(equipo.jugadores, i, jugador);
    }
    return equipo;}

Jugador crearJugador(int i){
```

```

    Jugador jugador;
    cout << "Ingrese el nombre del jugador nro" << i + 1 << ": ", cin >> jugador.nombre;
    cout << "Ingrese la calidad del jugador nro" << i + 1 << ": ", cin >> jugador.calidad;
    return jugador;
}

void calcularGoles(Equipo equipo1, Equipo equipo2, int size, int& golesEquipo1, int&
golesEquipo2){
    golesCalidad(equipo1, equipo2, size, golesEquipo1, golesEquipo2);
    golesMejorJugador(equipo1, equipo2, golesEquipo1, golesEquipo2);
    golesComparacion(equipo1, equipo2, size, golesEquipo1, golesEquipo2);
    golesPorSocio(equipo1, equipo2, golesEquipo1, golesEquipo2);
    golesPeorJugador(equipo1, equipo2, size, golesEquipo1, golesEquipo2);
}

void golesCalidad(Equipo equipo1, Equipo equipo2, int size, int& golesEquipo1, int&
golesEquipo2){
    int calidadEquipo1 = 0, calidadEquipo2 = 0;
    for(int i = 0; i < size; i++){
        calidadEquipo1 += equipo1.jugadores[i].calidad;
        calidadEquipo2 += equipo2.jugadores[i].calidad;
    }
    if (calidadEquipo1 > calidadEquipo2){
        golesEquipo1 += 2;
    } else if (calidadEquipo1 < calidadEquipo2){
        golesEquipo2 += 2;
    }
}

void golesMejorJugador(Equipo equipo1, Equipo equipo2, int& golesEquipo1, int& golesEquipo2){
    if(equipo1.jugadores[0].calidad > equipo2.jugadores[0].calidad){
        golesEquipo1 += 1;
    } else if(equipo1.jugadores[0].calidad < equipo2.jugadores[0].calidad){
        golesEquipo2 += 1;
    }
}

void golesComparacion(Equipo equipo1, Equipo equipo2, int size, int& golesEquipo1, int&
golesEquipo2){
    for(int i = 0; i < size; i++){
        if (equipo1.jugadores[i].calidad > equipo2.jugadores[i].calidad){
            golesEquipo1 += 1;
        } else if (equipo1.jugadores[i].calidad < equipo2.jugadores[i].calidad){
            golesEquipo2 += 1;
        }
    }
}

void golesPorSocio(Equipo equipo1, Equipo equipo2, int& golesEquipo1, int& golesEquipo2){
    if (equipo1.socios > equipo2.socios)
        golesEquipo1 += 1;
    else if (equipo1.socios < equipo2.socios)

```

```

        golesEquipo2 += 1;
    }

void golesPeorJugador(Equipo equipo1, Equipo equipo2, int size, int& golesEquipo1, int&
golesEquipo2) {
    if(equipo1.jugadores[size-1].calidad > equipo2.jugadores[size-1].calidad && golesEquipo2 >
0){
        golesEquipo2 -= 1;
    } else if(equipo1.jugadores[size-1].calidad < equipo2.jugadores[size-1].calidad &&
golesEquipo1 > 0){
        golesEquipo1 -= 1;
    }
}

void mostrarResultado(Equipo equipo1, Equipo equipo2, int golesEquipo1, int golesEquipo2){
    if (golesEquipo1 > golesEquipo2)
        cout << "Gano el equipo " << equipo1.nombre_equipo << " " << golesEquipo1 << " - " <<
golesEquipo2 << endl;
    else if (golesEquipo2 < golesEquipo1)
        cout << "Gano el equipo " << equipo2.nombre_equipo << " " << golesEquipo2 << " - " <<
golesEquipo1 << endl;
    else
        cout << "Empataron los equipos " << equipo1.nombre_equipo << " y " <<
equipo2.nombre_equipo << " " << golesEquipo1 << " - " << golesEquipo2 << endl;
}

```