

# Algoritmos y Estructuras de Datos

Oscar Meza

[oscarmeza@gmail.com](mailto:oscarmeza@gmail.com)

## Problema 1

Cree una carpeta llamada **prueba** y dentro, un archivo **prueba1.cpp**

Hacer un programa que imprima todas las permutaciones de 0,1 2, 3, 4  
Utilice for(...) o while



Problema 1

```
#include <iostream>
```

```
int main() {
    for (int i=0; i<=4; i++)
        for (int j=0; j<=4; j++)
            if (i!=j)
                for (int k=0; k<=4; k++)
                    if ((k!=i)&&(k!=j))
                        for (int l=0; l<=4; l++)
                            if ((l!=i)&&(l!=j)&&(l!=k))
                                for (int m=0; m<=4; m++)
                                    if ((m!=i)&&(m!=j)&&(m!=k)&&(m!=l))
                                        std::cout << i <<" "
                                            << j <<" "
                                            << k <<" "
                                            << l <<" "
                                            << m <<"\n";
}
```

## Problema 2

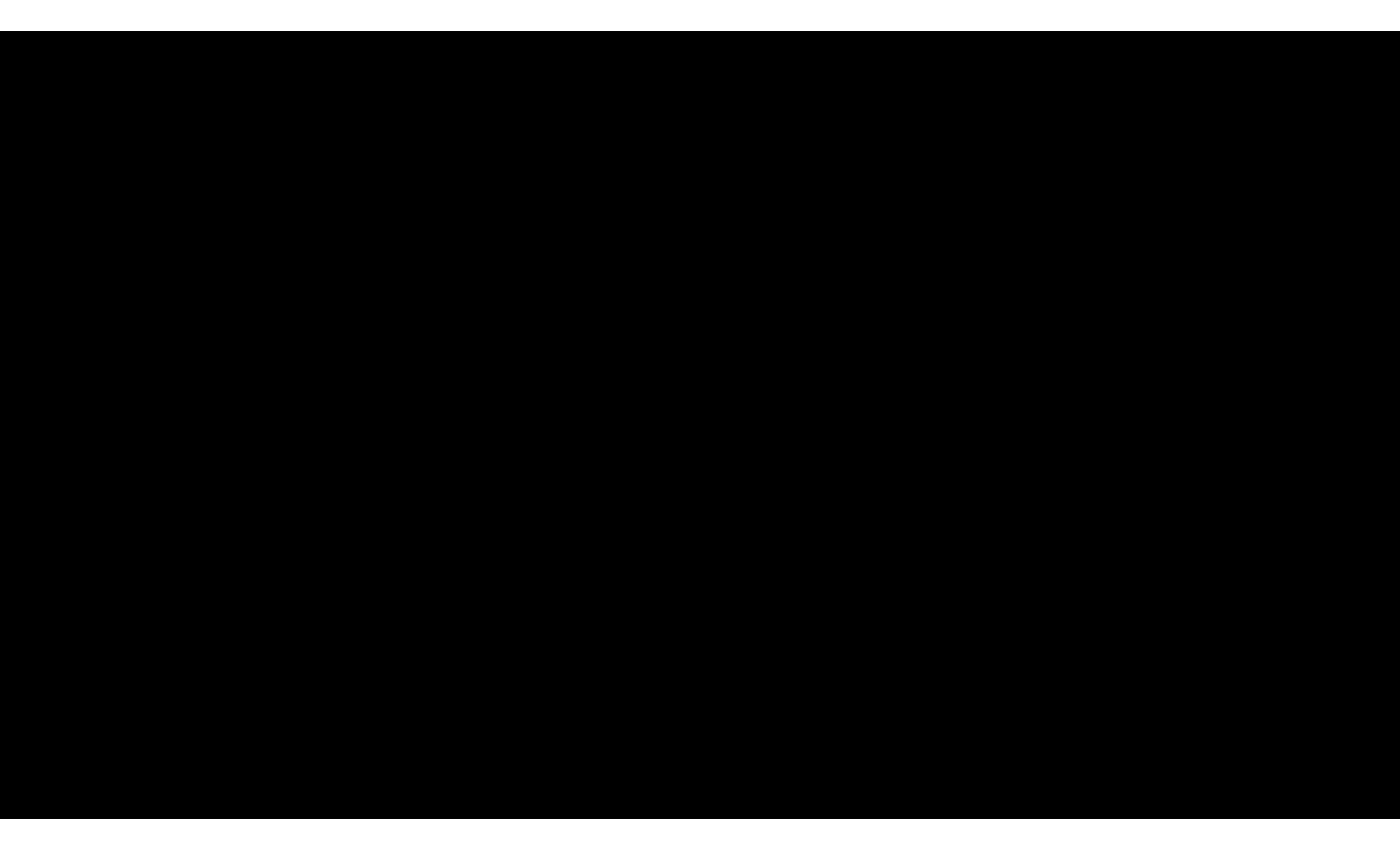
Crear un archivo **prueba2.cpp**

Hacer programa que imprima un rombo de asteriscos. N leído por pantalla representa el número de niveles del árbol(el no invertido):

Ejemplo n=4

```

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



## Problema 2:

```
#include <iostream>
```

```
int main(){
```

```
    int N;
```

```
    std::cout << "Coloque un número: ";
```

```
    std::cin>> N;
```

```
    for (int i=1; i<=N; i++){
```

```
        int blancos=N-i, asteriscos=2*(i-1)+1;
```

```
        for(int j=1;j<=blancos;j++)
```

```
            std::cout << " ";
```

```
        for(int j=1;j<=asteriscos;j++)
```

```
            std::cout << "*";
```

```
        std::cout <<'\n';
```

```
    }
```

```
for (int i=1; i<=N-1; i++){
```

```
    int blancos=i, asteriscos=2*(N-i)-1;
```

```
    for(int j=1;j<=blancos;j++)
```

```
        std::cout << " ";
```

```
    for(int j=1;j<=asteriscos;j++)
```

```
        std::cout << "*";
```

```
    std::cout <<'\n';
```

```
}
```

```
}
```

## Problema 3

Crear un archivo **prueba3.cpp**.

Hacer programa que invierta un vector leído por pantalla. Se debe crear una función que se le pase el vector como parámetro y devuelva el vector invertido. Se debe imprimir el vector invertido.

Si el vector tiene: 1 2 3 4

Lo convierte en: 4 3 2 1





Solución:

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

/*
 *  imprimir(v)
 *      Imprime en contenido de un vector de enteros
 *      v es el vector a imprimir
 */
void imprimir_vector(std::vector<int> v) {
    for (int elem : v)
        std::cout << elem << " ";
    std::cout << '\n';
}
```

```
/*
 * leer_vector( )
 *     Lee un vector de enteros
 *     no tiene parámetros
 *     Devuelve un vector de enteros leído
 */
std::vector<int> leer_vector(){    // devuelve un vector!!!
    int valor;
    std::vector<int> vector;
    std::cout << "Coloque los elementos "
        << "del vector y termine con la palabra fin: \n";
    while (std::cin>>valor) vector.push_back(valor);
    // Limpia el estado de error de la entrada
    // de datos por teclado
    std::cin.clear();
    // Vacía el buffer de entrada
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(),'\n');
    return vector;
}
```

```
std::vector<int> invertir(std::vector<int> vector){  
  
    for (int i =0 ; i <= (vector.size()-1)/2; i++)  
    {  
        int temporal;  
        temporal = vector[i];  
        vector[i]= vector[vector.size()-1-i];  
        vector[vector.size()-1-i] = temporal;  
    }  
    return vector;  
  
}
```

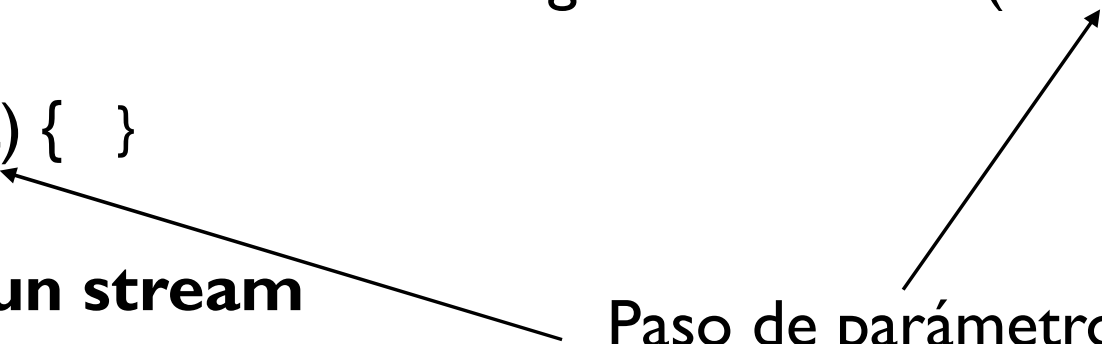
```
int main() {  
  
    std::vector<int> vector = leer_vector();  
    vector = invertir(vector);  
    imprimir_vector(vector);  
    return 0;  
  
}
```

## Dividir en palabras, una cadena de caracteres leída con espacios:

```
#include <iostream>
#include <string>
#include <vector>
#include <sstream>
using std::vector;
using std::string;
vector<string> Extraer(const string Text) { }
    vector<string> Words;
    std::stringstream ss(Text); // ss es un stream
    string Buf;
    while (ss >> Buf) // mientras existan palabras
        Words.push_back(Buf);
    return Words;
}
```

```
int main() {
    string linea ;
    std::cout << "Coloque una linea de texto: ";
    getline(std::cin, linea);
    vector<string> xx = Extraer(linea);
}
```

Paso de parámetro por  
VALOR



## Problema 4

Agregar al programa anterior una función “Buscar” que determina si una palabra dada (parámetro) está en un string leído por pantalla(parámetro). Utilizando la función Extraer dentro de Buscar.

En el programa principal leemos la línea con `getline`, leemos la palabra y llamamos a `Buscar` que llamará a `Extraer`





#### Problema 4

```
#include <iostream>
#include <string>
#include <vector>
#include <sstream>
using namespace std;
using std::string; // redundante
```

```
vector<string> Extraer(const string texto) ;
//necesario si colocamos
// primero Extraer
```

```
bool Buscar(const string texto, string palabra){
    vector<string> v = Extraer(texto);
    for(string w: v)
        if (w==palabra) return true;
    return false;
}
```

```
vector<string> Extraer(const string texto) {
    vector<string> palabras;
    stringstream ss(texto); // ss es un stream
    string palabra;

    while (ss >> palabra)
        palabras.push_back(palabra);

    return palabras;
}

int main() {

    string linea, palabra;
    cout << "Coloque una linea de texto: ";
    getline(cin,linea);
    cout << "Coloque palabra: ";
    cin >> palabra;
    if (Buscar(linea,palabra)) cout << palabra
<< " está\n";
    else cout << palabra << " no está\n";
}
```

## Problema 5:

Hacer un programa que lea dos matrices NxM , calcule la suma de las dos matrices e imprima las matrices y la suma. Les doy imprimir una matriz:

```
#include <iostream>
#include <vector>
#include <cstdlib> // para numeros random (que utilizaremos)
#include <iomanip> // para setw
using namespace std;
typedef vector<vector<double>> Matriz;
void imprimir_matriz(Matriz X){
    for (int i=0; i<X.size(); i++) {
        for (int j=0; j<X[i].size(); j++)
            cout << right<< setw(4) << X[i][j] ;
        cout << endl;
    }
}
```



Solucion:

```
Matriz sumar(Matriz X, Matriz Y){
    // X e Y deben tener las mismas
    // dimensiones NxM
    // con N > 0 y M > 0
    Matriz SUMA (X.size(),
        vector<double> (X[0].size()));
    for (int i=0; i<X.size(); i++ )
        for (int j=0; j<X[i].size(); j++ ){
            SUMA[i][j] = X[i][j] + Y[i][j];
        }
    return SUMA;
}
```

```
int main(){
    int N,M;
    // lee dos matrices y las suma e imprime el resultado
    cout << "Coloque las dimensiones \n"
        << "primero numero de filas y luego numero de columnas:";
    cin>> N >> M;
    Matriz A (N, vector<double> (M)),
        B (N, vector<double> (M)),
        SUMA (N, vector<double> (M));
    // genero dos matrices con numeros aleatorios del 0 al 99
    // Tambien las he podido leer por pantalla
    for (int i=0; i<A.size(); i++ )
        for (int j=0; j<A[i].size(); j++ ){
            A[i][j] = rand()%100;
            B[i][j] = rand()%100;
        }
}
```

```
cout << "Primera matriz:"<< endl;  
imprimir_matriz(A);  
cout << "Segunda matriz:"<< endl;  
imprimir_matriz(B);  
  
cout << "La suma es:"<< endl;  
imprimir_matriz(sumar(A,B));  
}
```

Algunos trucos y consejos....

Como poder leer y escribir caracteres del español por línea de comandos en windows:

```
#include <iostream>
#include <windows.h>
```

```
int main() {
    SetConsoleCP(1252); // Cambiar STDIN - Para máquinas Windows
                        // y poder leer e imprimir ñ á, etc
    SetConsoleOutputCP(1252); // Cambiar STDOUT - Para máquinas Windows

    char xx;
    std::cin >> xx;
    std::cout << xx << std::endl;
    std::cout << "Tamañño del vector = " << '\n';
}
```



## Podemos convertir números a string:

```
// CPP program to illustrate
// std::to_string
#include <bits/stdc++.h>
```

```
// Driver code
```

```
int main()
```

```
{
```

```
    int var1=16;
```

```
    // double a string
```

```
    std::string str1 = std::to_string(12.10);
```

```
    // entero a string
```

```
    std::string str2 = std::to_string(9999);
```

```
}
// Imprimir los strings
std::cout << str1 << '\n';
std::cout << str2 << '\n';
return 0;
```

En relación a la entrada y salida de datos por pantalla:

```
int x,y;  
std::cin >> x;  
std::cin >> y;
```

No hay por qué teclear enter para entrar **x** y luego **y**  
**Se pueden colocar en una misma línea del terminal y dar enter**

```
> 2 3
```

```
#include <iostream>
#include <limits>
int main() {
    int x;
    // Permite evitar errores de entrada por teclado
    // Por ejemplo que coloque una palabra en lugar de un entero
    std::cout << "Coloque el entero: ";
    // Permanece en el loop mientras no se introduzca un entero
    while (!(std::cin >> x)) {
        // Reporte el error
        std::cout << "Error de entrada, trate de nuevo: ";
        // Limpia el estado de error de la entrada de datos por teclado
        std::cin.clear();
        // Vacia el buffer de entrada
        std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
    }
    std::cout << "Usted tecleó" << x << '\n';
}
```

## Problema 6:

Haga una función que verifica si un string de caracteres es palíndrome, es decir, si invertimos la palabra que contiene el string resulta en la misma palabra original. Probar la función.

Ejemplo: el string contiene los caracteres A B C C B A, por lo que es palíndrome

Si contiene A B C no es palíndrome

Supongamos que tenemos el string: `std::string palabra;`

Introduzca la palabra por terminal.

## Problema 7

Hacer programa que imprima un rombo de asteriscos. N leído por pantalla representa el número de asteriscos que tiene el nivel con mas asteriscos:

Ejemplo n=7

```

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

n=6

```

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



### Problema 7:

```
#include <iostream>
```

```
int main() {
```

```
    int n {0};
```

```
    bool z {true};
```

```
    do {
```

```
        std::cout <<"coloque un numero "
                    <<"natural mayor que cero:";
```

```
        std::cin >> n;
```

```
        if (n<=0) break;
```

```
        for (int i = n/2 ; i>=0; i--) {
```

```
            // imprimir i blancos y luego n-2*i *
```

```
            for (int j = 1; j<=i; j++)
```

```
                std::cout <<' ';
```

```
            for (int j = 1; j<=n-2*i; j++)
```

```
                std::cout <<'*';
```

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

```
        }
```

```
        for (int i = 1 ; i<=n/2; i++) {
```

```
            // imprimir i blancos y luego n-2*i *
```

```
            for (int j = 1; j<=i; j++)
```

```
                std::cout <<' ';
```

```
            for (int j = 1; j<=n-2*i; j++)
```

```
                std::cout <<'*';
```

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

```
        }
```

```
    } while (z);
```

```
}
```

## Problema 8: Probar el programa siguiente...

El constructo switch:

```
#include <iostream>
```

```
int main() {
    int value;
    std::cout << "Coloque un numero entre 0 y 3: ";
    std::cin >> value;
    switch (value) {
        case 0:
            std::cout << "es cero";
            break;
        case 1:
            std::cout << "es uno";
            break;
        case 2:
            std::cout << "es dos";
            break;
```

```
        case 3:
            std::cout << "es tres";
            break;
        default:
            if (value < 0)
                std::cout << "Es negativo";
            else
                std::cout << "Es mayor que 3";
    }
```



### Problema 9: Probar el siguiente programa

```

*/
int main() {
    bool done = false; // Initially r
    do {
        switch (menu()) {
            case 'A': // Addition
            case 'a':
                get_input();
                add();
                report();
                break;
            case 'S': // Subtraction
            case 's':
                get_input();
                subtract();
                report();
                break;
            case 'P': // Print resul
            case 'p':
                report();
                break;
            case 'H': // Display hel
            case 'h':
                help_screen();
                break;
            case 'Q': // Quit the pr
            case 'q':
                done = true;
                break;
        }
    } while (!done);
}

```

```

char menu() {
    // Display a menu
    std::cout << "=== A)dd S)ubtract P)rint H)elp Q)uit ===\n";
    // Return the char entered by user
    char ch;
    std::cin >> ch;
    return ch;
}

```

## Ejercicios adicionales

- 1) Hacer una función que determine si un string es substring de otro.
- 2) Hacer una función que multiplique os matrices y devuelva la matriz resultado.
- 3) Hacer un programa que lea los coeficientes de un polinomio de grado N (lee los coeficientes de menor a mayor grado por pantalla) los guarda en un vector de double , y cree una función que evalúe el

## PREGUNTAS???

