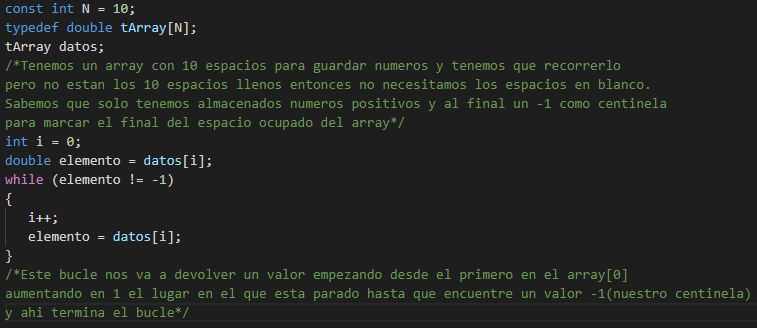
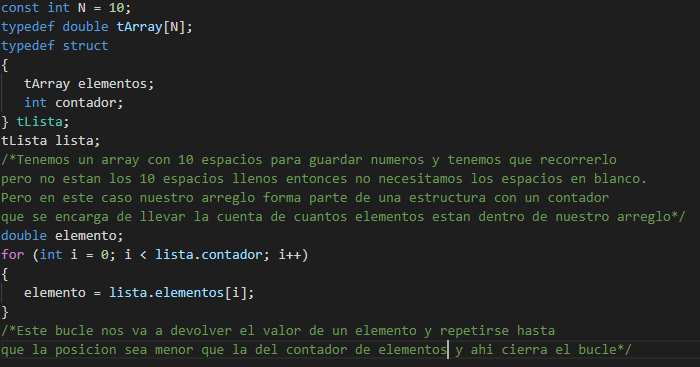
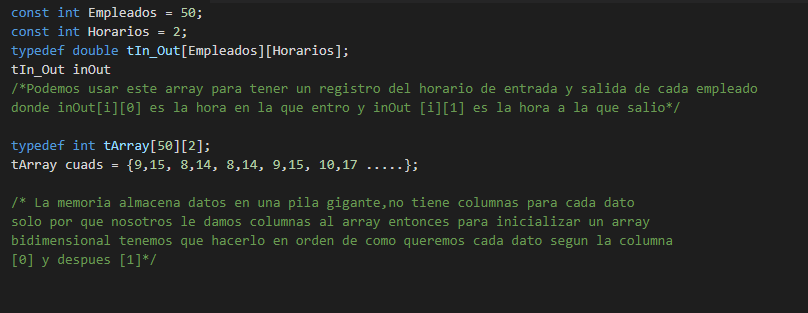
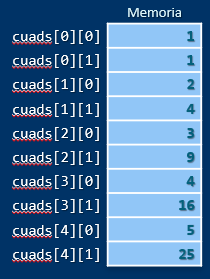
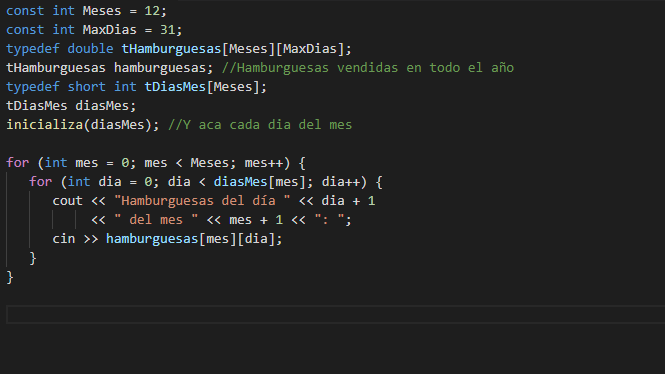
**Trabajo Práctico Nº 6 – Recorrido y búsqueda de arrays**Marco Teórico:  
  
1)

  
Guardamos un valor dato centinela para marcar el final de nuestro array, se usa cuando no tenemos nuestro array completamente ocupado y no necesitamos la información vacia. El centinela tiene que ser del mismo tipo de dato que el resto de los integrantes en el array pero tienen que tener ser un valor que no se debería encontrar ahí. Por ejemplo si tenemos un arreglo de números del 2 al 24 nuestro centinela podría ser 0,1 o cualquier valor mayor al 24.  
  
2)  
  
La diferencia con el bucle anterior es que con el centinela usábamos un WHILE por que iba a seguir dándonos valores SIEMPRE QUE NO SEA el numero centinela. Pero en este usamos un FOR para darnos un valor POR CADA espacio ocupado marcado por nuestro contador. Esto nos permite mucha mas libertad ya que el centinela nos limita a que tengamos un valor que no se pueda encontrar ni por error pero siendo del mismo tipo que el resto de los integrantes del array, con un contador en una estructura nos permite manejar diferentes tipos de arrays y también manejar otras estructuras con el contador.  
  
3)  
Un array multidimensional lo entiendo como un array que no es solo una fila de datos únicos si no una planilla con filas y columnas, similar a una hoja de Excel o una matriz. Esto nos facilita muchas formas de almacenar datos mas complicados como llevar una cuenta de las temperaturas en los días de la semana por un mes entero, o un registro de notas, etc.  
Aun que no estan limitados a solo dos dimensiones, por eso el nombre multidimensional, podemos darle las dimensiones que queramos en la definición tMatriz [2] [40] [6] [50];  
Pero necesitamos tantos índices como dimensiones cout << matrix [2] [40] [6] [50];  
  
4)  
  
  
Si pudiéramos ver como se guardan los datos en la memoria a medida que los inicializamos seria algo asi.   
  
Va a ir en orden de [0] [0] sumándole 1 primero a la columna y después a la siguiente fila para almacenar los datos entonces nosotros se los tenemos que inicializar en ese orden, con [ ] para cada dimensión del array.  
  
5)  
  
Cuando tenemos que recorrer un array multidimensional tenemos que hacer un bucle FOR para cada columna dentro del bucle FOR de cada línea. Como necesitamos todos los datos almacenados tenemos que recorrer primero cada una de las columnas para después poder pasar a la siguiente línea.

Marco Practico:  
  
A)

#include <iostream>

#include <string>

#include <fstream>

#include <iomanip>

using namespace std;

const int MaxProduct = 50;

const int STOCK = 2;

typedef int tStock[MaxProduct][STOCK];

typedef struct

{

    tStock stock;

    double pre;

} tProducto;

typedef tProducto tArray[MaxProduct];

typedef struct

{

    tArray producto;

    int cont = 0;

}tLista;

tLista lista;

tProducto producto;

ifstream leerLista(string path);

void mostrarLista();

void maxProducto();

void eliminarProducto();

ofstream guardarCambios(string path);

void closeFile(ifstream& file);

int main()

{

    int opcion;

    ifstream readFile = leerLista("ventas.txt");

    cout << "Menu lista de almacen" << endl;

    cout << "1.Ver la lista de productos" << endl;

    cout << "2.Para ver el producto de mayor valor en el listado" << endl;

    cout << "3.Para eliminar un producto del listado" << endl;

    cout << "0.Cerrar el programa" << endl;

    cin >> opcion;

    while (opcion != 0)

    {

        switch (opcion)

        {

            case 1:

            {

                mostrarLista();

            }

            break;

            case 2:

            {

                maxProducto();

            }

            break;

            case 3:

            {

                eliminarProducto();

            }

            break;

            default:

            {

                cout << "Opcion no disponible" << endl;

            }

        }

    cout << "Menu lista de almacen" << endl;

    cout << "1.Ver la lista de productos" << endl;

    cout << "2.Para ver el producto de mayor valor en el listado" << endl;

    cout << "3.Para eliminar un producto del listado" << endl;

    cout << "0.Cerrar el programa" << endl;

    cin >> opcion;

    }

    ofstream editFile = guardarCambios("ventas.txt");

    closeFile(readFile);

    return 0;

}

ifstream leerLista(string path)

{

    int codigo,unidades;

    double precio;

    int maximo = 0;

    ifstream archivo;

    archivo.open("ventas.txt");

    if (!archivo.is\_open())

    {

        cout << "No se pudo abrir el archivo " << endl;

    }

    else

    {

        archivo >> codigo >> precio >> unidades;

        while ((codigo != -1) && (lista.cont < 50))

        {

            lista.producto[maximo].stock[maximo][0] = codigo;

            lista.producto[maximo].pre = precio;

            lista.producto[maximo].stock[maximo][1] = unidades;

            maximo++;

            lista.cont++;

            archivo >> codigo >> precio >> unidades;

        }

    }

    return archivo;

}

void mostrarLista()

{

    int codigo,unidades;

    double precio;

    cout << "\t\t LISTA DE PRODUCTOS " << endl;

    for (int i = 0; i < lista.cont; i++)

    {

        codigo = lista.producto[i].stock[i][0];

        precio = lista.producto[i].pre;

        unidades = lista.producto[i].stock[i][1];

        cout << "Codigo: " << codigo << " Precio: " << precio << "$ Unidades disponibles: " << unidades << endl;

        cout << endl;

    }

}

void maxProducto()

{

    double valor = 0;

    int i, cod;

    for(int i = 0; i < lista.cont; i++)

    {

        if (((lista.producto[i].stock[i][1] \* lista.producto[i].pre) > valor && lista.producto[i].stock[i][0] != -1) )

        {

            valor = (lista.producto[i].stock[i][1] \* lista.producto[i].pre);

            cod = lista.producto[i].stock[i][0];

        }

    }

    cout << "El producto con mas valor en el almacen es" << endl;

    cout << "Codigo: " << cod << " Con un valor total de: " << valor << "$" << endl;

}

void eliminarProducto()

{

    int i, n, idAborrar;

    cout << "Ingrese el codigo del producto que desea eliminar: " << endl;

    cin >> idAborrar;

    for(i = 0; i <= lista.cont; i++)

    {

        if(lista.producto[i].stock[i][0] == idAborrar)

        {

            cout << "Producto eliminado " << endl;

            for(i ; i < lista.cont - 1; i++)

            {

                lista.producto[i].stock[i][0] = lista.producto[i + 1].stock[i + 1][0];

                lista.producto[i].stock[i][1] = lista.producto[i + 1].stock[i + 1][1];

                lista.producto[i].pre = lista.producto[i + 1].pre;

            }

            lista.cont--;

        }

    }

}

ofstream guardarCambios(string path)

{

    ofstream archivo;

    int i = 0;

    archivo.open("ventas.txt");

    while((lista.producto[i].stock[i][0] != -1) && (i < lista.cont))

    {

        archivo << lista.producto[i].stock[i][0] << " " << lista.producto[i].pre << " " << lista.producto[i].stock[i][1] << endl;

        i ++;

    }

    archivo << -1;

    return archivo;

}

void closeFile(ifstream& file)

{

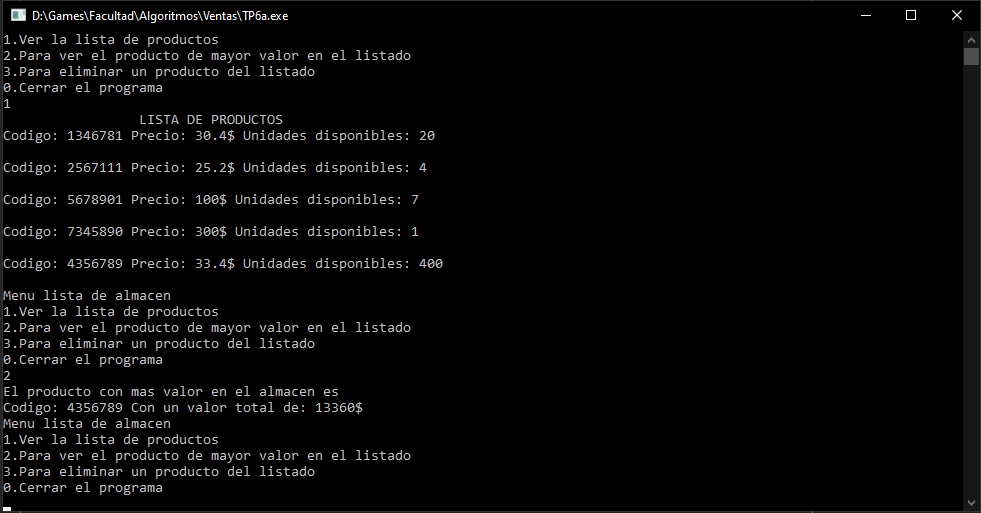
    if (file.is\_open())

    {

        file.close();

    }

}

  
  
  
  
B)

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

int main()

{

    const int Filas = 40;

    const int Columnas = 2;

    typedef string tOraciones [Filas][Columnas];

    tOraciones oraciones;

    string frase,palabra,largo;

    int contador = 0;

    cout << "Ingrese una frase: ";

    getline(cin,frase);

    for (char x : frase)

    {

        if ((x == ' ') || (x == ','))

        {

            oraciones[contador][0] = palabra;

            largo = (to\_string(palabra.length()));

            oraciones[contador][1] = largo;

            cout << "Palabra: " << oraciones[contador][0] << " /Cantidad de letras: " << oraciones[contador][1] << endl;

            contador++;

            palabra = "";

        }

        else {

            palabra = palabra + x;

        }

    }

    oraciones[contador][0] = palabra;

    largo = (to\_string(palabra.length()));

    oraciones[contador][1] = largo;

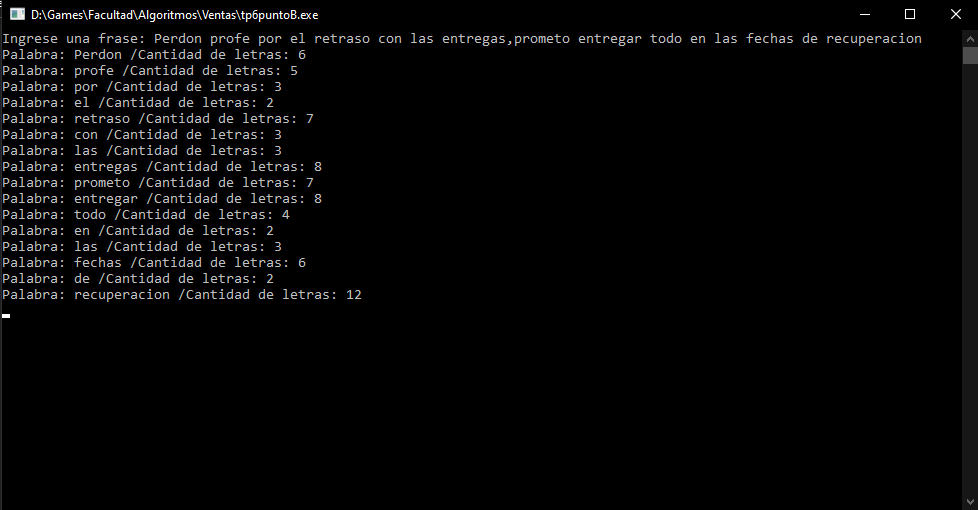
    cout << "Palabra: " << oraciones[contador][0] << " /Cantidad de letras: " << oraciones[contador][1] << endl;

    contador++;

    cin.get();

    return 0;

}



C)

#include<iostream>

using namespace std;

void sumarMatrices();

void multiplicarMatrices();

void transponerMatriz();

void puntoSilla();

int main()

{

    int opcion;

    cout << "Calculadora de matrices" << endl;

    cout << "1.Sumar matrices" << endl;

    cout << "2.Multiplicar matrices" << endl;

    cout << "3.Transponer matrices" << endl;

    cout << "4.Encontrar el punto de silla" << endl;

    cout << "0.Cerrar el programa" << endl;

    cin >> opcion;

    while (opcion != 0)

    {

        switch (opcion)

        {

            case 1:

            {

                sumarMatrices();

            }

            break;

            case 2:

            {

                multiplicarMatrices();

            }

            break;

            case 3:

            {

                transponerMatriz();

            }

            break;

            case 4:

            {

                puntoSilla();

            }

            break;

            default:

            {

                cout << "Opcion no disponible" << endl;

            }

        }

    cout << "Calculadora de Matrices" << endl;

    cout << "1.Sumar matrices" << endl;

    cout << "2.Multiplicar matrices" << endl;

    cout << "3.Transponer matrices" << endl;

    cout << "4.Encontrar el punto de silla" << endl;

    cout << "0.Cerrar el programa" << endl;

    cin >> opcion;

    }

    return 0;

}

void sumarMatrices()

{

    int filas, columnas, j, i;

    cout<<"Suma de matrices "<<endl;

    cout<<"Ingrese numero de filas: ";

    cin>>filas;

    cout<<"Ingrese numero de columnas: ";

    cin>>columnas;

    cout<<"\*\*\*\*\*\*\*\*\*\*\*"<<endl;

    int mat1[filas][columnas], mat2[filas][columnas];

    cout<<"Llenado de matriz 1"<<endl;

    for(i=0;i<filas;i++)

    {

        for(j=0;j<columnas;j++)

        {

        cout<<"Ingrese numero: ";

        cin>>mat1[i][j];

        }

    }

    cout<<"\n";

    cout<<"\*\*\*\*\*\*\*\*\*\*\*"<<endl;

    cout<<"Llenado de matriz 2"<<endl;

    for(i=0;i<filas;i++)

    {

        for(j=0;j<columnas;j++)

        {

            cout<<"Ingrese numero: ";

            cin>>mat2[i][j];

        }

    }

    cout<<"\n";

    cout<<"La matriz 1 es: "<<endl;

    for(int i=0;i<filas;i++)

    {

        for(int j=0;j<columnas;j++)

        {

            cout<<mat1[i][j]<<" ";

        }

        cout<<endl;

    }

    cout<<"\n";

    cout<<"La matriz dos es: "<<endl;

    for(i=0;i<filas;i++)

    {

        for(j=0;j<columnas;j++)

        {

            cout<<mat2[i][j]<<" ";

        }

        cout<<endl;

    }

    cout<<"\n";

    cout<<"La suma de las dos matrices es:"<<endl;

    for(i=0;i<filas;i++)

    {

        for(j=0;j<columnas;j++)

        {

            cout<<mat1[i][j]+mat2[i][j]<<" ";

        }

        cout<<endl;

    }

}

void multiplicarMatrices()

{

    //Pido medidas de las matrices

    int filaA, columA, filaB, columB, i, j, k;

    cout<<"Multiplicacion de matrices "<<endl;

    cout << "Ingrese el numero de filas de la primer matriz: " << endl;

    cin >> filaA;

    cout << "Ingrese el numero de columnas de la primer matriz: " <<endl;

    cin >> columA;

    cout << "Ingrese el numero de filas de la segunda matriz: " << endl;

    cin >> filaB;

    cout << "Ingrese el numero de columnas de la segunda matriz: " <<endl;

    cin >> columB;

    int mat1[filaA][columA], mat2[filaB][columB], producMat[filaA][columB];

    //Lleno la primer matriz con sus valores

    cout<<"\n";

    cout<<"\*\*\*\*\*\*\*\*\*\*\*"<<endl;

    cout<<"Llenado de matriz 1"<<endl;

    if(columA==filaB)

    {

        cout << "Introduzca los valores de la primer matriz: " << endl;

        for(i=0; i<filaA; i++)

        {

         for(j=0; j<columA; j++)

         {

            cout << "Ingrese numero: ";

            cin >> mat1[i][j];

         }

        }

        //Lleno la segunda matriz con sus valores

        cout<<"\n";

        cout<<"\*\*\*\*\*\*\*\*\*\*\*"<<endl;

        cout<<"Llenado de matriz 2"<<endl;

        cout << "Introduzca los valores de la segunda matriz: " << endl;

        for(i=0; i<filaB; i++)

        {

         for(j=0; j<columB; j++)

         {

            cout << "Ingrese numero: ";

            cin >> mat2[i][j];

         }

        }

        //Calculo el producto de las matrices

        for(i=0; i<filaA; i++)

        {

            for(j=0; j<columB; j++)

            {

                producMat[i][j]=0;

                for(k=0; k<columA; k++)

                {

                    producMat[i][j]+=mat1[i][k]\*mat2[k][j];

                }

            }

        }

        for(i=0; i<filaA; i++)

        {

            for(j=0; j<columB; j++)

            {

                cout << producMat[i][j]<<" ";

            }

            cout<<endl;

        }

    }

    else

    {

     cout << "La multiplicacion de matrices no se puede realizar,la cantidad de columnas de A no es igual " << endl;

     cout << "a la cantidad de filas de la matriz B,por favor revise las matrices y vuelva a intentarlo." << endl;

    }

}

void transponerMatriz()

{

    int filas, columnas, i, j;

    cout<<"Transponer una matriz "<<endl;

    cout << "Ingrese el numero de filas de la matriz: " << endl;

    cin >> filas;

    cout << "Ingrese el numero de columnas de la matriz: " <<endl;

    cin >> columnas;

    int matr[filas][columnas];

    cout << "Introduzca los valores de la matriz: " << endl;

    for(i=0; i<filas; i++)

    {

        for(j=0; j<columnas; j++)

        {

            cout << "Ingrese numero: ";

            cin >> matr[i][j];

        }

    }

    cout << "Matriz original: " << endl;

    for(i=0; i<filas; i++)

    {

        for(j=0; j<columnas; j++)

        {

            cout << matr[i][j] << " ";

        }

        cout << "\n";

    }

    cout << "Matriz transpuesta: " << endl;

    for(i=0; i<columnas; i++)

    {

        for(j=0; j<filas; j++)

        {

            cout << matr[j][i] << " ";

        }

        cout << "\n";

    }

}

void puntoSilla()

{

    int filas, colum, i, j;

    cout << "Ingrese el numero de filas de la matriz: " << endl;

    cin >> filas;

    cout << "Ingrese el numero de columnas de la matriz: " <<endl;

    cin >> colum;

    int matr[filas][colum], maximoFilas[filas], minimoColum[colum];

    cout << "Introduzca los valores de la matriz: " << endl;

    for(i=0; i<filas; i++)

    {

        for(j=0; j<colum; j++)

        {

            cout << "Ingrese numero: ";

            cin >> matr[i][j];

        }

    }

    for(i=0; i<filas; i++)

    {

        for(j=0; j<colum; j++)

        {

            if(matr[i][j] > maximoFilas[i])

            {

                maximoFilas[i] = matr[i][j];

            }

            if(matr[i][j] < minimoColum[j])

            {

                minimoColum[j] = matr[i][j];

            }

        }

    }

    for(i=0; i<filas; i++)

    {

        for(j=0; j<colum; j++)

        {

            if(maximoFilas[i] == minimoColum[j])

            {

                cout << "Punto de silla en: " << i << ", " << j << endl;

            }

        }

    }

}



