PRACTICA 7 FDP

EJERCICIO 1

```
#include <iostream>
#include <array>
using namespace std;
const int N=4;
const int M=5;
typedef array<int,M>TFila;
typedef array<TFila,N>TMatriz;

void Leerdatos (TMatriz& a) {
   cout << "Introduzca por filas una matriz " << N << " x " << M << " :" << endl;
   for (int i=0;i<N;i++) {
      for (int j=0;j<M;j++) {
         cin >> a[i][j];
      }
   }
}

void SaberMayor (const TMatriz& a) {
    int x=a[0][0],fil,col;
    for (int i=0;i<N;i++) {
      for (int j=0;j<M;j++) {
         if (x<a[i][j]) {
            x=a[i][j];
            fil=i+1;
            col=j+1;
         }
      }
   }
   cout << "El mayor de la matriz es: " << x << " que aparece en la posicion: [" << fil << "] [" << col <<
"]";
}


int main()
{
   TMatriz a;
   Leerdatos (a);
   SaberMayor (a);

   return 0;
}
```

EJERCICIO 2

```
#include <iostream>
#include <array>
using namespace std;
const int N=4;
typedef array<int,N>TFila;
typedef array<TFila,N>TMatriz;

void Leerdatos (TMatriz& a) {
   cout << "Introduzca por filas una matriz " << N << " x " << N << " :" << endl;
   for (int i=0;i<N;i++) {
      for (int j=0;j<N;j++) {
         cin >> a[i][j];
      }
   }
}

void SaberSim (TMatriz& a) {
   bool simetrica=true;
    for (int i=0;i<N;i++) {
      for (int j=0;j<N;j++) {
         if (a[j][i]!=a[i][j]) {
            simetrica=false;
         }
      }
   }
   if (simetrica==true) {
      cout << "SI es simetrica";
   } else {
      cout << "NO es simetrica";
   }
}

int main()
{
   TMatriz a;
   Leerdatos (a);
   SaberSim(a);

   return 0;
}
```

EJERCICIO 3

```
#include <iostream>
#include <array>
using namespace std;
const int N=5;
typedef array<int,N>TFila;
typedef array<TFila,N>TMatriz;

void Leerdatos (TMatriz& a) {
   cout << "Introduzca por filas una matriz " << N << " x " << N << " :" << endl;
   for (int i=0;i<N;i++) {
      for (int j=0;j<N;j++) {
         cin >> a[i][j];
      }
   }
}

unsigned sumaFila(const TMatriz& matriz, unsigned n, unsigned fila){
 unsigned suma = 0;
 for(unsigned i = 0; i < n; i++){
  suma+= matriz[fila][i];

 }

 return suma;
}

bool sumaFilas(const TMatriz& matriz, unsigned n, unsigned& ant1){
 bool sumasNoIguales= true;
 unsigned i = 1;
 ant1 = sumaFila(matriz, n, 0);
 while(sumasNoIguales== true && i < n){
  if(ant1 == sumaFila(matriz,n, i)){
   sumasNoIguales = true;
  }else{
   sumasNoIguales = false;
  }
  i++;
 }
 return sumasNoIguales;
}

unsigned sumaColumna(const TMatriz& matriz, unsigned n, unsigned columna){
 unsigned suma = 0;
  for(unsigned i = 0; i < n; i++){
   suma+= matriz[i][columna];
```

```
  }

  return suma;
}

bool sumaColumnas(const TMatriz& matriz, unsigned n, unsigned& ant2){
 bool a = true;
 unsigned i = 1;
 ant2 = sumaColumna(matriz, n, 0);
 while(a== true && i < n){
  if(ant2 == sumaColumna(matriz,n, i)){
   a = true;
  }else{
   a = false;
  }
  i++;
 }
 return a;
}

unsigned sumaDiagonalP(const TMatriz& matriz, unsigned n){
 unsigned suma = 0;
 for(unsigned i = 0; i< n; i++){
   for(unsigned j = 0; j< n; j++){
    if(i==j){
     suma += matriz[i][j];
    }
   }
  }
 return suma;
}

unsigned sumaDiagonalS(const TMatriz& matriz, unsigned n){
 unsigned suma = 0;
 unsigned j = n-1;
 for(unsigned i = 0; i < n; i++){
  suma += matriz[i][j];
  j--;
 }
 return suma;
}

bool Es_Magica(const TMatriz& A, unsigned n, unsigned ant1, unsigned ant2){
 bool es=false;
 bool sumaFilasIgual, sumaColumnasIgual;
 sumaFilasIgual = sumaFilas(A, n, ant1);
 sumaColumnasIgual = sumaColumnas(A, n,ant2);
```

```
 if(ant1 == ant2){
 if(sumaFilasIgual== true && sumaColumnasIgual== true){

  if(sumaDiagonalP(A, n) == sumaDiagonalS(A,n) ){
    es = true;
   }
 }
 }
 return es;
}

int main() {
 TMatriz a;
 Leerdatos(a);
 int ant1=0, ant2=0;
 bool es;
 es = Es_Magica(a,N,ant1,ant2);
 cout << endl;

 if(es){
  cout << "Es un cuadro magico";
   } else {
    cout << "No es un cuadrado magico";
  }

 return 0;
}
```

EJERCICIO 4

```
#include <iostream>
#include <array>
using namespace std;
const int N=5;
typedef array<int,N>TFila;
typedef array<TFila,N>TMatriz;

void Iniciar(TMatriz& a, int k, int fil, int col){
    cout << "BIENVENIDOS AL CUADRADO MAGICO" << endl;
    for (int i=0;i<N;i++) {
        for (int j=0;j<N;j++) {
            a[i][j]=0;
            a[fil][col]=k;
        }
    }
   /* for (int i=0;i<N;i++) {
        for (int j=0;j<N;j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    } */
}

void Secuencia(TMatriz& a, int& k, int& fil, int& col, int cont) {
   while (k<cont) {
      k++;
      fil--;
      col--;
    if (fil==-1) {
        fil=N-1;
    }
    if (col==-1) {
        col=N-1;
    }
    if ((a[fil][col]!=0)) {
        fil=fil+2;
        col++;
    }
    a[fil][col]=k;
    if ((fil==0)&&(col==0)) {
      fil=2;
      col=1;
   }
   }
}
```

```
void Mostrar (const TMatriz& a) {
    for (int i=0;i<N;i++) {
        for (int j=0;j<N;j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
}

int main()
{
    TMatriz a;
    int k=1,fil=0,col=N/2;
    int cont=N*N;
    Iniciar(a,k,fil,col);
    Secuencia(a,k,fil,col,cont);
    Mostrar(a);

    return 0;
}
```

EJERCICIO 5

```
#include <iostream>
#include <array>
using namespace std;
const int MAX=10;
typedef array<int,MAX>TVector;
typedef array<bool,MAX>TBool;

void Rellenar (TBool& a) {
   for (int i:a) {
      a[i]=true;
   }
}

void Leerdatos (TVector& v1) {
   cout << "Introduzca una sucesion de " << MAX << " numeros naturales: ";
   for (int i=0;i<MAX;i++) {
      cin >> v1[i];
      if (v1[i]<=0) {
         cout << "ERROR: Numero no natural";
      }
   }
}

int CONT(TVector& v1,int x) {
   int cont=0;
   for (int j=0;j<MAX;j++) {
      if (x==v1[j]) {
         cont++;
      }
   }
   return cont;
}

int MAYOR (const TVector& v1, TBool& a) {
   int x=v1[0];
   for (int i=0;i<MAX;i++) {
      if ((v1[i]>x)&&(a[i]==true)) {
         x=v1[i];
         //cout << x  << "   " << i+1 << endl << endl << endl;
      }
   }
   return x;
}

void Secuencia1 (TVector& v1, TBool& a) {
   int k=0;
```

```
    while (k<MAX) {
     int cont;
     int x=MAYOR(v1,a);
     for (int i=0;i<MAX;i++) {
       if ((x==v1[i])&&(a[i]==false)) {
          cont=CONT(v1,x);
       }
     }
     cout << x;
     for (int i=0;i<MAX;i++) {
       if (x==v1[i]) {
          a[i]=false;
       }
     }
     if (cont>1) {
     cout << " aparece " << cont << " veces, en posiciones ";
     } else {
        cout << " se repite " << cont << " vez, en posicion ";
     }
     for (int i=0;i<MAX;i++) {
       if (x==v1[i]) {
          cout << i+1 << " ";
       }
     }
     cout << ".";
     cout << endl;
     k+=cont;
}
}



int main()
{
    TVector v1;
    TBool a;
    Rellenar (a);
    Leerdatos(v1);
    Secuencia1(v1,a);

    return 0;
}
```

EJERCICIO 6ª

```
#include <iostream>
#include <array>
using namespace std;
const int MAXC=15,MAXP=10;
typedef array<int,MAXP>TVoto;
typedef array<string,MAXP>TArrayP;
typedef array<int,MAXC>TFila;
typedef array<TFila,MAXP>TMatriz;
typedef array<bool,MAXC>TFilaB;
typedef array<TFilaB,MAXP>TMatrizB;
struct TCargos{
   int num;
   TFila carg;
};
struct TVotos {
   int num;
   TVoto votos;
};
struct TNombres {
   int num;
   TArrayP part;
};
struct TCargoElecto{
   TArrayP nom;
   TVoto num;
};

void Leerdatos (TCargos& c, TNombres& p, TMatriz& a, TVotos& v) {
   cout << "Introduzca el Numero de Cargos: ";
   cin >> c.num;
   cout << "Introduzca el Numero de Partidos: ";
   cin >> p.num;
   v.num = p.num;
 cout << "Introduzca el Nombre y Numero de Votos por Partido:" << endl;
 for (int i=0;i<p.num;i++) {
   cout << "Partido: " << i+1 << ": ";
   cin >> p.part[i];
   cin >> v.votos[i];
    }
   /* for (int i=0;i<p.num;i++) {
      cout << p.part[i] << "-->" << v.votos[i] << endl;
} */
}

void Rellenar (TMatrizB& b, TNombres& p, TCargos& c, TCargoElecto& ce) {
    for (int i=0;i<p.num;i++) {
```

```
          for (int j=0;j<c.num;j++) {
        b[i][j]=true;
       }
       }
       for (int i=0;i<p.num;i++) {
        ce.num[i]=0;
        ce.nom[i]=p.part[i];
       }
}

int SaberMayor (const TMatriz& a,  TMatrizB& b, const TNombres& p, const TCargos& c,
TCargoElecto& ce) {
   int mayor=a[0][c.num-1];
   for (int i=0;i<p.num;i++) {
      for (int j=0;j<c.num;j++) {
         if ((mayor<=a[i][j])&&(b[i][j]==true)) {
            mayor=a[i][j];
         }
      }
    }
    for (int i=0;i<p.num;i++) {
      for (int j=0;j<c.num;j++) {
         if (a[i][j]==mayor){
            b[i][j]=false;
            ce.num[i]++;
         }
      }
    }

    return mayor;
}

void Secuencia1 (const TCargos& c, const TNombres& p, TMatriz& a, const TVotos& v, TCargoElecto&
ce, TMatrizB& b) {
   int mayor;
   int k=0;
   for (int i=0;i<p.num;i++) {
          for (int j=1;j<c.num;j++) {
        a[i][0]=v.votos[i];
        a[i][j]=(a[i][0]/(j+1));
       }
       }

   while (k<c.num) {
        mayor=SaberMayor(a,b,p,c,ce);
    //   cout << k+1 << " " << mayor << endl;
      k++;
}
```

```
  /* cout << endl << endl << endl;
   for (int i=0;i<p.num;i++) {
      for (int j=0;j<c.num;j++) {
         cout << a[i][j] << "  ";
      }
      cout << endl;
   }  */
}

void Salida (const TCargoElecto& ce, const TNombres& p, const TCargos& c) {
   cout << "Los Cargos Electos son: " << endl;
   for (int i=0;i<c.num;i++) {
      if (ce.num[i]>0) {
      cout << ce.nom[i] << " " << ce.num[i] << endl;
     }
  }
}


int main()
{
   TCargos c;
   TVotos v;
   TMatriz a;
   TNombres p;
   TMatrizB b;
   TCargoElecto ce;
   Leerdatos (c,p,a,v);
   Rellenar(b,p,c,ce);
   Secuencia1 (c,p,a,v,ce,b);
   Salida(ce,p,c);

   return 0;
}
```

EJERCICIO 6B

```
#include <iostream>
#include <array>
using namespace std;
const int FILAS = 5;
const int COLUMNAS = 5;
typedef array<int, COLUMNAS> TFilaSup;
typedef array<TFilaSup, FILAS> Superficie;
typedef array<char, COLUMNAS> TFilaLav;
typedef array<TFilaLav, FILAS> Lava;

void Iniciar (Lava& lava, int fil, int col) {
    for (int i = 0; i < FILAS; i++) {
      for (int j = 0; j < COLUMNAS; j++) {
      lava[i][j]=' ';
      }
     }
     lava[fil][col]='*';
}

void flujoDeLava(const Superficie& sup, int fil, int col, Lava& lava){
   int altura=sup[fil][col];
  // cout << endl << altura << endl;
    for (int i=0;i<FILAS;i++) {
       for (int j=0;j<COLUMNAS;j++) {
          if
((lava[i][j]!='*')&&(((col==j)&&((fil==i+1)||(i==fil+1)))||((fil==i)&&((col==j+1)||(j==col+1))))&&(sup[i][j
]<altura)) {
             lava[i][j]='*';
             fil=i;
             col=j;
             flujoDeLava(sup,fil,col,lava);
          }
       }
     }
}

int main()
{
  Superficie sup;
  Lava lava;
  int fil,col;
  cout << "Introduzca superficie (matriz de naturales " << FILAS << "x" << COLUMNAS << "):\n";
  for (int i = 0; i < FILAS; i++) {
     for (int j = 0; j < COLUMNAS; j++) {
     cin >> sup[i][j];
```

```
     }
    }
    cout << "Introduzca punto de crater (fila y columna):\n";
    cin >> fil >> col;
    Iniciar(lava,fil,col);
    flujoDeLava(sup,fil,col,lava);
    cout << "El recorrido de la lava es:\n";
    for (int i = 0; i < FILAS; i++) {
     for (int j = 0; j < COLUMNAS; j++) {
     cout << lava[i][j];
     }
     cout << endl;
    }
    return 0;
}
```