**Solucionario del Examen Sustitutorio de Algoritmia y Estructura de Datos (SI205)**

**Pregunta 1:**

#include <iostream>

#include <fstream>

using namespace std;

float m[5][5];

//Función que lee del archivo

void Leer(float m[5][5]){

ifstream ent;

int i,j;

ent.open("MATRIZ.TXT");

if(!ent){

cout<<"Error de apertura...";

}

else{

for(i=0; i<5; i++){

for(j=0; j<5; j++){

ent>>m[i][j];

}

}

ent.close();

}

}

//Función recursiva que determinar el máximo (M) o mínimo (m)

float Valor(float m[5][5], int fila, int columna, float mn, char c){

if((fila>=0 && fila<5)&&(columna>=0 && columna<5)){

switch (c){

case 'M': if(m[fila][columna]>mn){

mn=m[fila][columna];

}

break;

case 'm': if(m[fila][columna]<mn){

mn=m[fila][columna];

}

};

if(columna==4){

columna=0;

fila++;

}

else{

columna++;

}

return Valor(m,fila,columna,mn,c);

}

return(mn);

}

//Función recursiva que Presenta las posiciones del máximo o mínimo

void Presenta(float m[5][5], int fila, int columna, float mn){

if((fila>=0 && fila<5)&&(columna>=0 && columna<5)){

if(m[fila][columna]==mn){

cout<<fila<<" ,"<<columna<<endl;

}

if(columna==4){

columna=0;

fila++;

}

else{

columna++;

}

Presenta(m,fila,columna,mn);

}

}

int main(){

Leer(m);

cout<<"Mayor elemento es:"<<Valor(m,0,0,m[0][0],'M')<<endl;

cout<<"Aparece en las posiciones:"<<endl;

Presenta(m,0,0,Valor(m,0,0,m[0][0],'M'));

cout<<"Menorr elemento es:"<<Valor(m,0,0,m[0][0],'m')<<endl;

cout<<"Aparece en las posiciones:"<<endl;

Presenta(m,0,0,Valor(m,0,0,m[0][0],'m'));

return 0;

}

**Pregunta 2:**

//Crear un Nodo para el Distrito

//Crear un Nodo para la Provincia, que tenga un campo puntero a un Nodo del tipo Distrito.

//Crear un Nodo para el Departamento que tenga un campo puntero a un Nodo Provincia.

//Crear los recorridos que permitan determinar la parte a) y b).

**Pregunta 3:**

#include <iostream>

#include <string.h>

using namespace std;

typedef char Cadena[201];

typedef char Cadena2[31];

int main(){

Cadena texto;

int i,j,l,q,m,np;

bool existe;

cout<<"Ingrese la oracion: "; gets(texto);

q=strlen(texto);

m=0;

i=0;

while(i<q){

while(i<q && texto[i]==' '){

i++;

}

if(i<q){

m++;

}

while(i<q && texto[i]!=' '){

i++;

}

}

Cadena2 p[m];

Cadena2 pal;

int ap[m];

np=0;

i=0;

int my=0;

while(i<q){

while(i<q && texto[i]==' '){

i++;

}

strcpy(pal,"");

j=0;

while(i<q && texto[i]!=' '){

pal[j]=texto[i];

j++;

i++;

}

pal[j]='\0';

if(strlen(pal)>0){

existe=false;

l=0;

while(l<np && !existe){

if(strcmp(pal,p[l])==0){

existe=true;

ap[l]++;

}

else{

l++;

}

}

if(!existe){

strcpy(p[np],pal);

ap[np]=1;

np++;

}

}

if(ap[l]>my){

my=ap[l];

}

}

cout<<"Las palabras que mas repiten son: "<<endl;

for(i=0; i<np; i++){

if(ap[i]==(my)){

cout<<p[i]<<endl;

}

}

return 0;

}