#include <iostream>

#include <iomanip>

#include <cstdlib>

#include <string>

#include <cstring>

#include <algorithm>

#include <fstream>

#include <sstream>

#include <queue>

#include <stack>

#define INP "input.INP"

#define OUT "output.OUT"

using namespace std;

typedef int item;

typedef struct GRAPH

{

char \*name; // ten cac dinh

item \*\*G; // ma tran trong so

int n; // so phan tu cua do thi

} Graph;

void input\_file(Graph &Gr);// lay du lieu tu file

void input\_B\_E(Graph Gr, int &a, int &b); //nhap vao dinh dau va cuoi

void output\_file(Graph Gr);//Xuat ket qua tu file ra

void Menu(int &select); //menu chon thuat toan

int Dijkstra(Graph Gr, int a, int b);//thuat toan Dijkstra

int number\_or\_char(Graph Gr); //nhap vao kiem tra la ky tu hay so va tra ve vi tri cua dinh trong do thi

item tongthiethai(Graph Gr); //tong quang duong di cua moi dinh (thay the cho vo cung trong ma tran trong so)

string convert\_to\_string(int number);//chuyen so number sang chuoi

int floyd (Graph Gr, int a, int b);

int main()

{

Graph Gr;

input\_file(Gr);

int a, b, \*P, i, select = 1;

output\_file(Gr);

while (select)

{

Menu(select);

switch (select)

{

case 1:

{

cout<<endl<<"-----Thuat toan Dijkstra-----"<<endl;

input\_B\_E(Gr, a, b);

Dijkstra(Gr, a, b);

system("pause");

break;

}

case 2:

{

cout<<endl<<"-----Thuat toan Floy-----"<<endl;

input\_B\_E(Gr, a, b);

floyd (Gr, a, b);

system("pause");

break;

}

}

if (select == 3) break;

}

system("pause");

return 0;

}

void input\_file(Graph &Gr)

{

ifstream inp(INP);

if (inp == NULL)

{

cout<<"No found file input";

return;

}

inp >> Gr.n ;

Gr.name = new char [Gr.n];

for (int i=0; i<Gr.n; i++)

inp >> Gr.name[i];

Gr.G = new int \*[Gr.n];

for (int i=0; i<Gr.n; i++)

{

Gr.G[i] = new int [Gr.n];

for (int j=0; j<Gr.n; j++)

inp >> Gr.G[i][j];

}

inp.close();

}

void input\_B\_E(Graph Gr, int &a, int &b)

{

a = b = 0;

cout<<endl<<"Cac dinh danh so tu 1 den "<<Gr.n<<".Hoac tu "<<Gr.name[0]<<" den "<<Gr.name[Gr.n-1]<<endl;

cout<<"Nhap dinh bat dau : ";

while (a<1 || a> Gr.n)

{

cin>>a;

if (a<1 || a> Gr.n)

cout<<"Khong hop le ! \nNhap lai dinh bat dau : ";

}

cout<<"Nhap dinh ket thuc : ";

while (b<1 || b> Gr.n)

{

cin>>b;

if (b<1 || b> Gr.n)

cout<<"Khong hop le ! \nNhap lai dinh ket thuc : ";

}

a -- ;

b -- ;

}

void output\_file(Graph Gr)

{

//ofstream out(OUT);

fstream out;

out.open(OUT, ios::out|ios::trunc);

cout<<"Ma tran ke cua do thi"<<endl<<endl;

out<<"Ma tran ke cua do thi"<<endl<<endl;

for (int i=0; i<Gr.n; i++)

{

cout<<setw(2)<<Gr.name[i];

out<<setw(2)<<Gr.name[i];

}

out<<endl<<endl;

cout<<endl<<endl;

for (int i=0; i<Gr.n; i++)

{

for (int j=0; j<Gr.n; j++)

{

cout<<setw(2)<<Gr.G[i][j];

out<<setw(2)<<Gr.G[i][j];

}

cout<<endl;

out<<endl;

}

out.close();

}

//tong quang duong di cua moi dinh (thay the cho vo cung trong ma tran trong so)

item tongthiethai(Graph Gr)

{

item sum = 0;

for (int i=0; i<Gr.n; i++)

for (int j=0; j<Gr.n; j++)

sum += Gr.G[i][j];

return sum;

}

void Menu(int &select)

{

cout<<endl<<"Moi ban chon thuat toan :"<<endl;

cout<<"1: Thuat toan Dijkstra"<<endl;

cout<<"2: Thuat toan Floyd"<<endl;

cout<<"3: Thoat !"<<endl;

cin >> select;

}

int Dijkstra(Graph Gr, int a, int b)

{

fstream out;

out.open(OUT, ios::out|ios::app);

out<<endl<<"\*\*\*\*\*"<<endl;

// Len[i] - Gia tri nho nhat tu a -> i. Len1 danh dau do dai.

int Len[Gr.n];

int S[Gr.n];//Danh dau dinh thuoc danh sach dac biet

int P[Gr.n];//truy vet

int max = tongthiethai(Gr);

fill (Len,Len+Gr.n,max); //Gan duong di ban dau = vo cung

fill (P,P+Gr.n,a);

fill (S,S+Gr.n,0); //Danh sach dac biet

Len[a] = 0; // khoi tao do dai tu a->a = 0

int i = a, dem = 0, space = 10;

//in ra hang tieu de

out<<setw(space/2)<<"TT |";

for (int i=0; i<Gr.n; i++)

{

char s[100];

sprintf(s,"%d%c%c%c",i+1,'(',Gr.name[i],')');

out<<setw(space)<<s;

}

out <<endl;

for (int i=0; i< (space/2) + Gr.n\*10; i++)

out<<"-";

out<<endl;

//ket thuc in ra hang tieu de

//while S<>V

for (int k=0; k<Gr.n; k++)

{

dem ++;

char \*s = new char [100];

char vocung = '~' , gach[10] = " - ";

out<<setw(space/2-2)<<dem<<" |";

//tim do dai ngan nhat trong cac dinh

for (i=0; i<Gr.n; i++) // tim v thuoc (V-S) va Len[v] < vo cung

if (!S[i] && Len[i] != max)

break;

for (int j = i+1 ; j<Gr.n; j++) // tim dinh co Len min

if (!S[j] && Len[j] < Len[i])

i = j;

S[i] = 1;

//----------In ra gia tri moi lan lap------------

if (dem > 0)

for (int j=0; j<Gr.n; j++)

{

char temp[100];

strcpy(s," ");

if (dem >1 && j != i && S[j])

sprintf(s,"%c",'-');

else

{

if (j == i || (dem == 1 && j == a))

strcat(s,"\*");

strcat(s,"[");

if ( j != i && !S[j] && Len[j] == max)

sprintf(temp,"%c,",vocung);

else

sprintf(temp,"%d,",Len[j]);

strcat(s,temp);

if (j!=a && k==0)

sprintf(temp, "%c", vocung);

else sprintf(temp, "%d", P[j]+1);

strcat(s,temp);

strcat(s,"]");

}

out<<setw(space)<<s;

}

//----------Ket thuc In ra gia tri moi lan lap------------

//--------Tinh do dai tu dinh dang xet toi cac dinh tiep

for (int j = 0; j<Gr.n; j++) //thay doi do dai neu co

{

if (!S[j] && Gr.G[i][j])

if (Len[i] + Gr.G[i][j] < Len[j])

{

Len[j] = Len[i] + Gr.G[i][j];

P[j] = i; //truy vet

}

}

out<<endl;

}

//Ket luan duong di

out<<endl<<"Do dai ngan nhat cua duong di tu "<<a+1<<"("<<Gr.name[a]<<")"

<<" den "<<b+1<<"("<<Gr.name[b]<<")"<<" la "<<Len[b]<<endl;

out<<"Qua trinh duong di: ";

i = b;

char \*s, \*temp;

s = new char [Gr.n\*10];

temp = new char [10];

sprintf(s,"%d",i+1);

while (i != a)

{

sprintf(temp,"%s"," --> ");

strcpy(s,strcat(temp,s));

sprintf(temp,"%d",P[i] +1);

strcpy(s,strcat(temp,s));

i = P[i];

}

out<<s<<endl;

cout<<"Hoan thanh ! Mo file output.out de xem ket qua !";

out.close();

return Len[b];

}

int floyd (Graph Gr, int a, int b)

{

fstream out;

out.open(OUT, ios::out|ios::app);

int max = tongthiethai(Gr);

int A[Gr.n][Gr.n], P[Gr.n][Gr.n];

for (int i=0; i<Gr.n; i++)

for (int j=0; j<Gr.n; j++)

{

if (Gr.G[i][j])

A[i][j] = Gr.G[i][j];

else A[i][j] = max;

P[i][j] = -1;

}

for (int k=0; k<Gr.n; k++)

{

out<<endl<<"Buoc thu "<<k<<endl;

out<<setw(2\*Gr.n)<<"A"<<setw(15+4\*Gr.n)<<"P"<<endl;

for (int i=0; i<Gr.n; i++)

{

for (int j=0; j<Gr.n; j++)

{

char \*temp = new char [50];

if (A[i][j] == max)

sprintf(temp,"%c",'~');

else

sprintf(temp,"%d",A[i][j]);

out<<setw(4)<<temp;

}

out<<setw(15)<<" ";

for (int j=0; j<Gr.n; j++)

out<<setw(4)<<P[i][j] + 1;

out<<endl;

}

for (int i=0; i<Gr.n; i++)

for (int j=0; j<Gr.n; j++)

if (A[i][j] > A[i][k] + A[k][j])

{

A[i][j] = A[i][k] + A[k][j];

P[i][j] = k ;

}

}

out<<endl<<"Do dai ngan nhat cua duong di tu "<<a+1<<"("<<Gr.name[a]<<")"<<" den "<<b+1<<"("<<Gr.name[b]<<")"<<" la "<<A[a][b]<<endl;

out<<"Qua trinh duong di: ";

//truy vet

char \*s, \*temp;

s = new char [Gr.n\*10];

temp = new char [10];

stack <item> S1;

stack <item> S2;

S1.push(a); //danh sach nap cac dinh vao

S1.push(b); //danh sach xuat cac dinh ra

int dich, tg;

while (!S1.empty())

{

dich = S1.top(); //dich = phan tu dau tien

S1.pop(); // dua phan tu do ra

S2.push(dich); //cho vao danh sach xuat

if (!S1.empty()) //trong khi S1 ko rong thi tiep tuc tim cac dinh

{

tg = S1.top();

while (P[tg][dich] != -1) //tim cac dinh di tu tg den dich

{

S1.push(P[tg][dich]);

tg = S1.top();

}

}

}

sprintf(s,"%d",S2.top()+1);

S2.pop();

while (!S2.empty())

{

sprintf(temp,"%s%d"," --> ",S2.top()+1);

strcat(s,temp);

S2.pop();

}

out<<s<<endl;

cout<<"Hoan thanh ! Mo file output.out de xem ket qua !";

out.close();

return A[a][b];

}