#include <iostream>

#include <stdio.h>

#include <fstream>

#include <sstream>

using namespace std;

#define MAX 20

//DSLK sử dụng cho QUEUE và DANH SÁCH KỀ

struct Node

{

int info;

Node \*link;

};

Node \*First[MAX]; //DS các liên kết

int n; //Số đỉnh của đồ thị

char vertex[MAX]; //Tên đỉnh

Node \*sp;

Node \*front, \*rear;

////QUEUE

void Init\_Queue()

{

front = NULL;

rear = NULL;

}

int isEmptyQ()

{

if(front == NULL)

return 1;

return 0;

}

void PushQ(int x)

{

Node \*p = new Node;

p->info = x;

p->link = NULL;

if(rear == NULL)

front = p;

else

rear->link = p;

rear = p;

}

int PopQ(int &x)

{

if(front != NULL)

{

Node \*p = front;

front = p->link;

x = p->info;

if(front == NULL)

rear = NULL;

delete p;

return 1;

}

return 0;

}

///End QUEUE

//7.1:Nhập danh sách kề của đồ thị

void Init\_Graph()

{

n = 0;

}

void Insert\_Last(Node \*&first, Node \*p)

{

if(first == NULL)

first = p;

else

{

Node \*q = first;

while(q->link != NULL)

q = q->link;

q->link = p;

}

}

//7.2: Xuất danh sách kề ra màn hình

void Output\_Graph()

{

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

{

cout << "Dinh "<< i <<" ("<< vertex[i] <<"): ";

Node \*p = First[i];

while(p != NULL)

{

cout << p->info << " ";

p = p->link;

}

cout << endl;

}

}

void Output(int a[], int n)

{

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

cout << vertex[ a[i] ] << " ";

}

//7.3: Duyệt đồ thị theo chiều rộng

int C[100], bfs[100];

int nbfs = 0;

void InitC()

{

for(int i=0; i<n; i++) //n là số đỉnh của đồ thị

C[i] = 1;

}

void BFS(int v) //v là đỉnh bắt đầu

{

int p;

Node \*w;

PushQ(v);

C[v] = 0;

while(front != NULL)

{

PopQ(p);

bfs[nbfs] = p;

nbfs++;

while(w != NULL)

{

if(C[w->info])

{

PushQ(w->info);

C[w->info] = 0;

}

w = w->link;

}

}

}