Meno: Michal Korbela
trieda: sexta
škola: Gymnázium J.J. Bánovce
úloha č.3

Tak asi najoptimálnejšie riešenie je zapamätať si chodby v matici susednosti.
Potom môžu nastať 3 prípady:

1. cieľová miestnosť je súčasne aj východiskovou
   na to potrebujeme konštantne času
2. z východiskovej miestnosti sa dá dostať do cieľovej pomocou 1 chodby – to len skontrolujeme, či susedí cieľová s východiskovou miestnosťou - potrebujeme konštantne času
3. do cieľovej sa dá dostať pomocou 2 chodieb
   mohli by sme skúsiť BFS, ale to je v tomto prípade hodne pomalé, preto vysúšame urobiť niečo takéto:
   zapamätáme si susedov východiskovej miestnosti a potom ich porovnávame so susedmi cieľovej miestnosti. Ak sa nájde zhoda, tak sme vyhrali.
   Na toto potrebujeme lineárne veľa času, pretože potrebujeme skontrolovať všetkých susedov východiskovej a všetkých susedov cieľovej miesnosti. A tých je dokopy 2N, čiže potrebujeme lineárne veľa času od počtu miestností.

Takže sme zistili, že v najhoršom prípade potrebujeme na jednu otázku lineárne veľa času, takže na všetky otázky to bude O(q.n), kde n je počet miestností a q je počet otázok.

Čo sa týka pamäťovej zložitosti, tak tá je O(n^2), pretože si musíme zapamätať maticu susedností. Susedov, ktorých si musíme zapamätať v 3. prípade, tak tých je lineárny počet od n, takže pamäťová zložitosť zostáva stále kvadratická.

```
#include <iostream>
#include <string>
#include <stdlib.h>
#include <sstream>
#include <math.h>
#include <stdio.h>
#include <vector>
using namespace std;



bool pole[15000][15000];                    //matica susedností



int main(){
int n,m,k;

cin>>n>>m>>k;

for(int i=0; i<m; i++){
int p1,p2;
```

```
cin>>p1>>p2;
pole[p1][p2]=true;
pole[p2][p1]=true;                          //načítame vstup do matice susedností
}



for(int i=0; i<k; i++)                      //odpovedáme otázky
{

int p,q,ok=0;
cin>>p>>q;
if(p==q)ok=1;                               //ak je východisková zároveň cieľovou
if(pole[p][q]==true) ok=1;                  //ak východisková susedí s cieľovou
if(ok==0){
vector <int> V;
for(int i=1; i<=n; i++){
if(pole[p][i]==true) V.push_back(i);        //zistíme si susedov východiskovej miestnosti
}

for(int i=n; i>0; i—){                      //hľadáme susedov odzadu, aby sme ich mohli
rovno porovnávať s tými vo vectore
if(pole[q][i]==true){
int d=0;
while(ok==0 && !V.empty()){

if(V.back()>=i){
d=V.back();
V.pop_back();                               //ak je sused príliš veľký, nikdy sa k nemu už
nedostaneme, tak preto ho odstránime
}
else break;
if(d==i){ ok=1;}                            //ak sme našli zhodu s susedoch

}

}
if(ok==1) break;                            //ok končíme

}


}
if(ok==1) cout<<"Ano"<<endl;                //vypíšeme
else cout<<"Nie"<<endl;
}
```

}