

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**ОТЧЕТ по практической работе**

**по дисциплине “Методы программирования”**

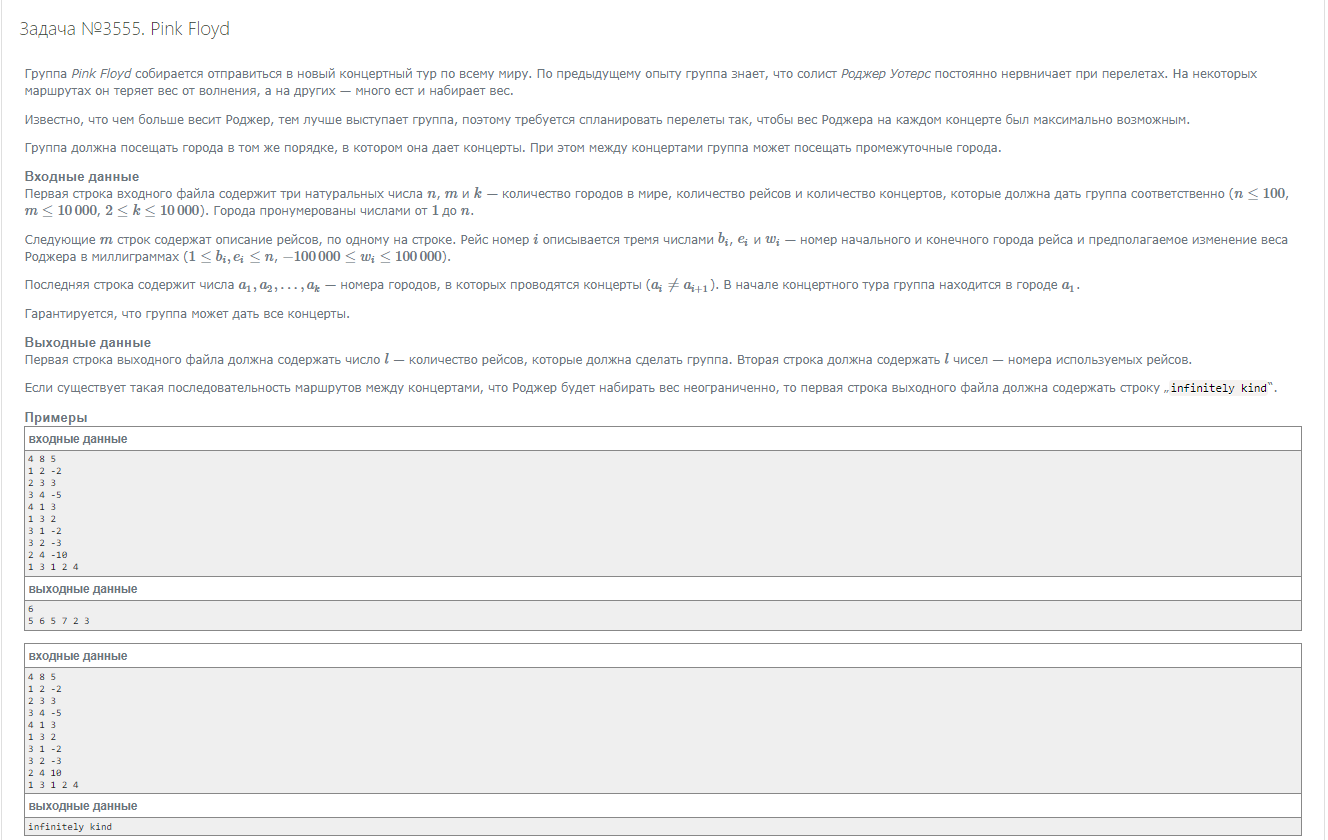
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автор |  | | |  | Шеременко С.Е. | | | |
|  | (подпись, дата) | | |  |  | | | |
| Обозначение | 10.05.01.550000.000 О | | Группа | | | | ВКБ31 |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | | | | |
| Профиль | Компьютерная безопасность | | | | | | |
| Преподаватель |  | | |  | | Савельев В.А. | |
|  | (подпись, дата) | | |  | |  | |

г. Ростов-на-Дону

2021 год

**Лабораторная работа №4**

Задание



КОД

#include <iostream>

#include <vector>

#include <map>

using namespace std;

int ret(vector<vector<int>>& from, int u, int v, vector<int>& path)

{

if (u == v)

{

return 0;

}

if (from[u][v] == u || from[u][v] == v)

{

path.push\_back(v);

return 0;

}

ret(from, u, from[u][v], path);

ret(from, from[u][v], v, path);

return 0;

}

int main()

{

const int inf = -1e9;

bool cycle = false;

int n, m, k;

cin >> n >> m >> k;

vector<vector<int>> graph(n, vector<int>(n, inf)), from(n, vector<int>(n, -1));

vector<int> a(k + 1, 0), path, ans;

map<pair<int, int>, int> flights;

path.push\_back(0);

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

{

int b, e, w;

cin >> b >> e >> w;

b--, e--;

graph[b][e] = w;

flights[{b, e}] = i;

from[b][e] = b;

}

for (int i = 1; i <= k; i++)

{

cin >> a[i];

a[i]--;

}

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

{

graph[i][i] = 0;

}

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

{

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

{

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

{

if (graph[i][j] < graph[i][K] + graph[K][j])

{

graph[i][j] = graph[i][K] + graph[K][j];

from[i][j] = K;

}

}

}

}

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

{

if (graph[a[i]][a[i]] > 0)

{

cycle = true;

}

}

if (cycle)

{

cout << "infinitely kind";

}

else

{

for (int j = 1; j <= k; j++)

{

ret(from, a[j - 1], a[j], path);

}

for (int i = 1; i < path.size(); i++)

{

ans.push\_back(flights[{path[i - 1], path[i]}] + 1);

}

cout << ans.size() << endl;

for (auto el : ans)

{

cout << el << " ";

}

}

}

Результат

