**M. Ali. Arslan**

**19F-0348**

**Lab-12**

**Task # 01**

**Task # 02**

#include <iostream>

#include<vector>

#include<queue>

using namespace std;

class Graph

{

public:

vector<vector<int>> g;

int v = 4;

int visitedvertex(int x, vector<int>& path)

{

int size = path.size();

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

{

if (path[i] == x)

{

return 0;

}

return 1;

}

}

void printpath(vector<int>& path)

{

int size = path.size();

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

{

cout << path[i] << " ";

}

cout << endl;

}

void searchpaths(vector<vector<int> >& g, int source, int des, int v)

{

queue<vector<int> > q;

vector<int> path;

path.push\_back(source);

q.push(path);

while (!q.empty()) {

path = q.front();

q.pop();

int last = path[path.size() - 1];

if (last == des)

{

printpath(path);

}

for (int i = 0; i < g[last].size(); i++)

{

if (visitedvertex(g[last][i], path))

{

vector<int> newpath(path);

newpath.push\_back(g[last][i]);

q.push(newpath);

}

}

}

}

};

int main()

{

Graph h;

h.g.resize(6);

h.g[1].push\_back(3);

h.g[1].push\_back(2);

h.g[2].push\_back(4);

h.g[2].push\_back(3);

h.g[4].push\_back(0);

h.g[0].push\_back(3);

h.g[0].push\_back(5);

h.g[5].push\_back(3);

int source = 2, des = 3;

cout << "All the paths from " << source << " to " << des << " are: " << endl;

h.searchpaths(h.g, source, des, h.v);

return 0;

}

