







№6

#include <iostream>

#include <vector>

#include <fstream>

#include <queue>

#define white 0

#define black 1

**using** **namespace** std;

**int** main()

{

    ifstream fin("pathbge1.in");

    ofstream fout("pathbge1.out");

**int** n, m;

    fin >> n >> m;

    vector <queue<**int**>> dot(n);

    vector <**bool**> color(n, white);

    vector <**int**> min\_path(n, 0);

**int** left, right;

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

    {

        fin >> left >> right;

        left--;

        right--;

        dot[left].push(right);

        dot[right].push(left);

    }

    queue <**int**> dot\_friend;

    dot\_friend.push(0);

**while** (!dot\_friend.empty())

    {

**int** date = dot\_friend.front();

        dot\_friend.pop();

        color[date] = black;

**while** (!dot[date].empty())

        {

**if** (color[dot[date].front()] == white)

            {

**int** new\_date = dot[date].front();

                min\_path[new\_date] = min\_path[date] + 1;

                dot\_friend.push(new\_date);

                color[new\_date] = black;

            }

            dot[date].pop();

        }

    }

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

        fout << min\_path[i] << " ";

}

№7

Алгоритм тот же, что и в предыдущей задаче.

№8

#include <iostream>

#include <vector>

#include <queue>

#include <string>

using namespace std;

int main()

{

freopen("knight1.in", "r", stdin);

freopen("knight1.out", "w", stdout);

string aa, bb;

cin >> aa >> bb;

int a1 = aa[0] - 'a', a2 = aa[1] - '1';

int b1 = bb[0] - 'a', b2 = bb[1] - '1';

vector < vector < int>> v(64, vector<int>(64, 0));

for (int i = 0; i < 64; i++)

for (int j = 0; j < 64; j++)

if ((abs(i - j) == 6 || abs(i - j) == 10 || abs(i - j) == 15 || abs(i - j) == 17) && (abs(i % 8 - j % 8) <= 2)) v[i][j] = 1;

queue<int> q;

vector<int> f(64, 1e9);

q.push(a1 \* 8 + a2);

f[a1 \* 8 + a2] = 0;

while (!q.empty())

{

int w = q.front();

q.pop();

for (int i = 0; i < 64; i++)

{

if (v[w][i] == 1 && f[i] == 1e9)

{

q.push(i);

f[i] = w;

}

}

}

vector<int> s;

for (int i = b1 \* 8 + b2; i != a1 \* 8 + a2; i = f[i]) s.push\_back(i);

cout << aa << endl;

for (int i = s.size() - 1; i >= 0; i--) cout << char(s[i] / 8 + 'a') << char(s[i] % 8 + '1') << endl;

}

