#include <iostream>

#include <windows.h>

using namespace std;

const int k = 5; // chess board size

int ar[k][k];

const int shift\_count = 8; //варианты смещения

int rollbacks = 0;

COORD shift[]

{

{ 1, -2 }, { 2, -1 }, { 2, 1 }, { 1, 2 },

{ -1, 2 }, { -2, 1 }, { -2, -1 }, { -1, -2 }

};

void show\_state()

{

COORD info{ 0, 1 };

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(h, info);

SetConsoleTextAttribute(h, 10);

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

{

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

{

cout << "\t" << ar[y][x];

}

cout << "\n\n";

}

}

void message()

{

COORD msg{ 45, 1 };

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(h, msg);

SetConsoleTextAttribute(h, 5);

cout << " Message: ";

}

void rtrn()

{

COORD rtrn{ 45, 3 };

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(h, rtrn);

SetConsoleTextAttribute(h, 9);

cout << " Quantity of returns: ";

}

bool pony(int x, int y, int step)

{

show\_state(); // перемещение коня на доске 5х5 пошагово с интервалом в 1 секунду.

Sleep(1000);

ar[y][x] = step; // первый шаг и тд

if (step >= k \* k) // если массив заполнен до конца

return true;

for (int i = 0; i < shift\_count; i++) // проверка границ нашей доски

{

if ((x + shift[i].X >= 0) && (x + shift[i].X < k) && // лево-право

(y + shift[i].Y >= 0) && (y + shift[i].Y < k) && // верх-низ

(ar[y + shift[i].Y][x + shift[i].X] == 0)) // если равно нулю, то место свободно и ставим цифру

{

if (pony(x + shift[i].X, y + shift[i].Y, step + 1)) // проверка следующего шага

{

return true;

}

else

{

rollbacks++; // счётчик "откатов" (заходов в тупик)

show\_state();

COORD msg{ 55, 1 };

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(h, msg);

SetConsoleTextAttribute(h, 12);

cout << " Deadlock"; // сообщение о заходе в тупик

Sleep(500);

SetConsoleCursorPosition(h, msg); // стирание сообщения

cout << " ";

COORD rtrn{ 67, 3 };

SetConsoleCursorPosition(h, rtrn);

SetConsoleTextAttribute(h, 14);

cout << rollbacks; // количество "откатов" (заходов в тупик)

}

}

}

ar[y][x] = 0;

show\_state(); // показ "откатов" (заходов в тупик)

Sleep(1000);

return false;

}

int main()

{

system("title Horse Move");

int x, y;

cout << "Put position, two numbers from 0 to " << k - 1 << ":\n";

cin >> x >> y;

if (x < 0 || x > k - 1) x = 0;

if (y < 0 || y > k - 1) y = 0;

system("cls");

message();

rtrn();

pony(x, y, 1);

// results

show\_state();

system("pause");

}