代码：

#include <iostream>

using namespace std;

class xy {

private:

int x;

int y;

public:

xy() {

this->x = -1;

this->y = -1;

}

xy(int m,int n) {

this->x = m;

this->y = n;

}

const int Getx() const {

return x;

}

const int Gety() const {

return y;

}

};

ostream& operator << (ostream& o, const xy& num)

{

o << "[" << num.Getx() << "," << num.Gety() << "]";

return o;

}

int main()

{

int map[15][15] = { { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 },

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

{ 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1 },

{ 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1 },

{ 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1 },

{ 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1 },

{ 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1 },

{ 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1 },

{ 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1 },

{ 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1 },

{ 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1 },

{ 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1 },

{ 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1 },

{ 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1 },

{ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 } };

xy\* Stack = new xy[225];

int n = 0;

for (int i = 0; i <= 14; i++) {

for (int j = 0; j <= 14; j++) {

if (map[i][j] == 2) {

Stack[0] = xy(i, j);

i = 15;

break;

}

}

}

while (n>=0&&n<=225) {

int i = Stack[n].Getx();

int j = Stack[n].Gety();

if (map[i][j - 1] == 3) {

break;

}

else {

if (map[i][j + 1] == 3) {

break;

}

else {

if (map[i - 1][j] == 3) {

break;

}

else {

if (map[i + 1][j] == 3) {

break;

}

}

}

}

if (map[i][j - 1] == 0) {

Stack[n+1] = xy(i, j - 1);

n++;

map[i][j-1] = 1;

}

else {

if (map[i][j + 1] == 0) {

Stack[n+1] = xy(i, j + 1);

n++;

map[i][j+1] = 1;

}

else {

if (map[i-1][j] == 0) {

Stack[n+1] = xy(i - 1, j);

n++;

map[i-1][j] = 1;

}

else {

if (map[i + 1][j] == 0) {

Stack[n + 1] = xy(i + 1, j);

n++;

map[i+1][j] = 1;

}

else {

Stack[n] = xy(-1, -1);

n--;

}

}

}

}

}

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

cout << Stack[i] << " ";

}

delete[]Stack;

}

结果：

输出正确路线的x，y坐标

