**《动感迷宫》实习报告**

题目：编制一个求解迷宫通路的程序

班级：19184115

专业：计算机科学与技术（人工智能与大数据）创新实验班

姓名：洪智豪 学号：19195216

完成日期：2020.10.1

1. **需求分析**
2. 自定义迷宫大小

1) 规定用户输入的迷宫长宽范围为[5,15]

2) 输入异常时，则报告相应信息，然后返回输入界面，直至用户输入合法为止

1. 以随机生成或文件导入的形式生成迷宫
2. 起点规定为（1，0），终点规定为（M-2，N-1）
3. 随机生成迷宫时，路与墙的初始比例约为7：3，如果一条路四周是墙，打通这些墙；如果一个条路四周都是路，让这条路变为墙
4. 文件导入迷宫时，打开根目录下的input.txt中的迷宫

第一行两个数组代表行与列，其余1表示墙，0表示路

1. 当导入的迷宫不存在通路，没有解，周围一圈除起点和终点外还有其他路，长与宽不符合规定时，提示用户文件的标准形式，然后返回主界面。
2. 动态显示走迷宫的过程
3. 用实心方块 ‘■’ 表示墙；空心方块 ‘□’表示路； ‘X’ 表示死路，用箭头表示实时走迷宫的方向
4. 通过延时刷新迷宫来展示走迷宫的实时状态
5. 以（x1,y1）-> (x2,y2) 的形式输出走过的路径
6. 输出迷宫所有可能的路径
7. 输出所有的走法与路径长度
8. 输出路径的最大值可自定义
9. 按回车显示下一条道路
10. 输出迷宫的所有可能的最短路径
11. 输出迷宫的所有可能的最短路径
12. 输出路径的最大值可自定义
13. 输出最短路径的长度，以及最短路径的条数
14. 自定义遍历顺序
15. 用户按输入要求键入遍历顺序
16. 输入异常时，则报告相应信息，然后返回输入界面，直至用户输入合法为止
17. 用户自己通过键入WSAD来走迷宫
18. 设定时间限制，时间上限为从起点走到终点的最短步数+4
19. 通过用户的输入来动态刷新迷宫
20. 用户的当前位置标记为：○，终点标记为：●，走过的路标记为×
21. 当超时或者陷入死胡同时，输出游戏失败，然后返回主界面
22. 在规定时限内抵达终点后，根据用户的用时来评分，然后返回主界面

1. **概要设计**
2. 数据结构设计
3. 栈的抽象数据类型定义：

typedef struct SqStack

{

SElemType \*base;

SElemType \*top;

int stacksize;

} SqStack;

\*base：

1. 获取栈顶元素

Status GetTop(SqStack \*s, SElemType \*e)

1. 构造一个空栈

Status GetTop(SqStack \*s, SElemType \*e)

1. 元素出栈

Status Pop(SqStack \*s, SElemType \*e)

1. 元素入栈

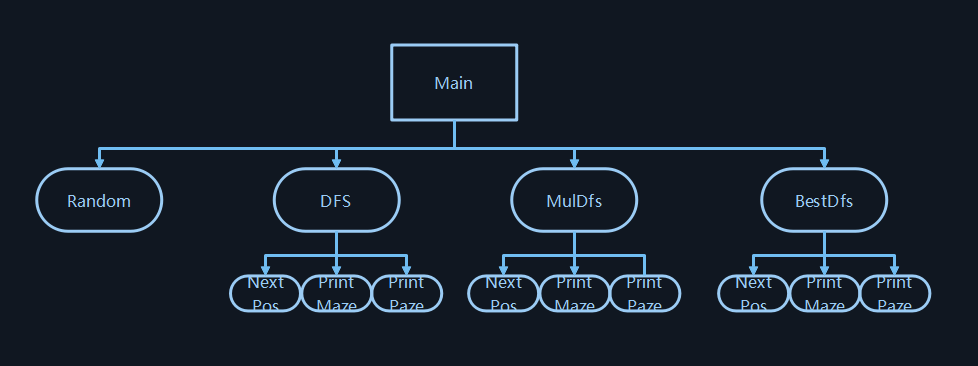
Status Push(SqStack(\*s), SElemType e)

1. 判断栈是否为空

Status StackEmpty(SqStack s)

1. 清空栈中所有元素

void StackClear(SqStack \*s)

1. 模块设计
2. 模块结构图：
3. 函数说明

项目中包含了以下函数：

1. void Random()

//随机生成迷宫的函数

1. Status Judge(PosType e)

//判断e这一位置是否可走  
e：节点

1. PosType NextPos(PosType e,int dir)

//确定下一步的节点

e:节点; dir:代表方向的标记

1. Void PrintPath(SqStack \*s)

//输出路径

s:栈

1. Void PrintMaze()

//打印迷宫

1. Void Dfs(PosType start,PosType end,SqStack \*s,int sta)

//sta=0时，检查是否可通;sta=1时，深搜并动态显示路径

start:起点;end:终点; s:栈; sta:状态

1. Void MulDfs(PosType start,PosType end,SqStack \*s)

//深度搜索所有路径

start:起点;end:终点; s:栈

1. Void BestDfs(PosType start,PosType end,SqStack \*s,int len)

//深度搜索所有最短路径

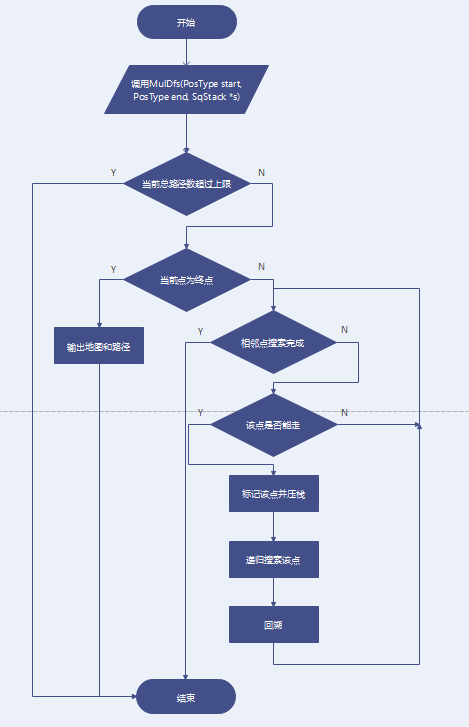
start:起点;end:终点; s:栈;路径长度: len

1. Int main()

//主函数 打印用户界面 同时通过调用其他函数完成用户的需求

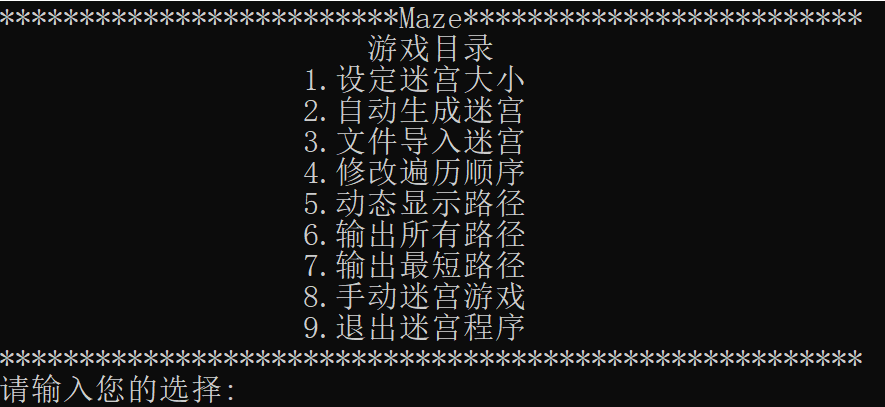
1. **详细设计**

**程序流程图**



1. **用户手册**

1）主界面



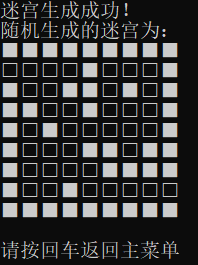
2）设定迷宫大小

C:\Users\Riter\AppData\Roaming\Tencent\Users\767641495\QQ\WinTemp\RichOle\O3~PR])NZG]FO@~P(YOVT4C.png

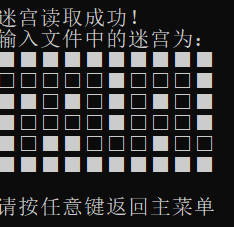
C:\Users\Riter\AppData\Roaming\Tencent\Users\767641495\QQ\WinTemp\RichOle\L0$5A`T5LGMC9(Z]2W)V5Y5.png

C:\Users\Riter\AppData\Roaming\Tencent\Users\767641495\QQ\WinTemp\RichOle\28KXBOAL$T3EVF}CS9U3`BX.png

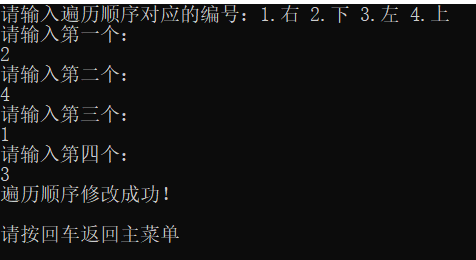
3）自动生成迷宫



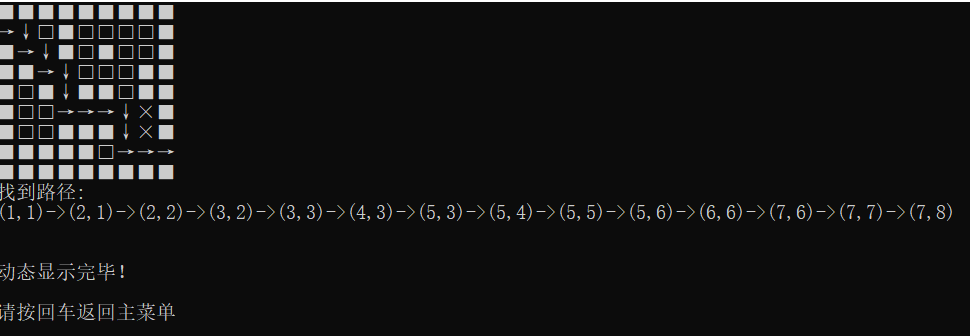
4）文件导入迷宫



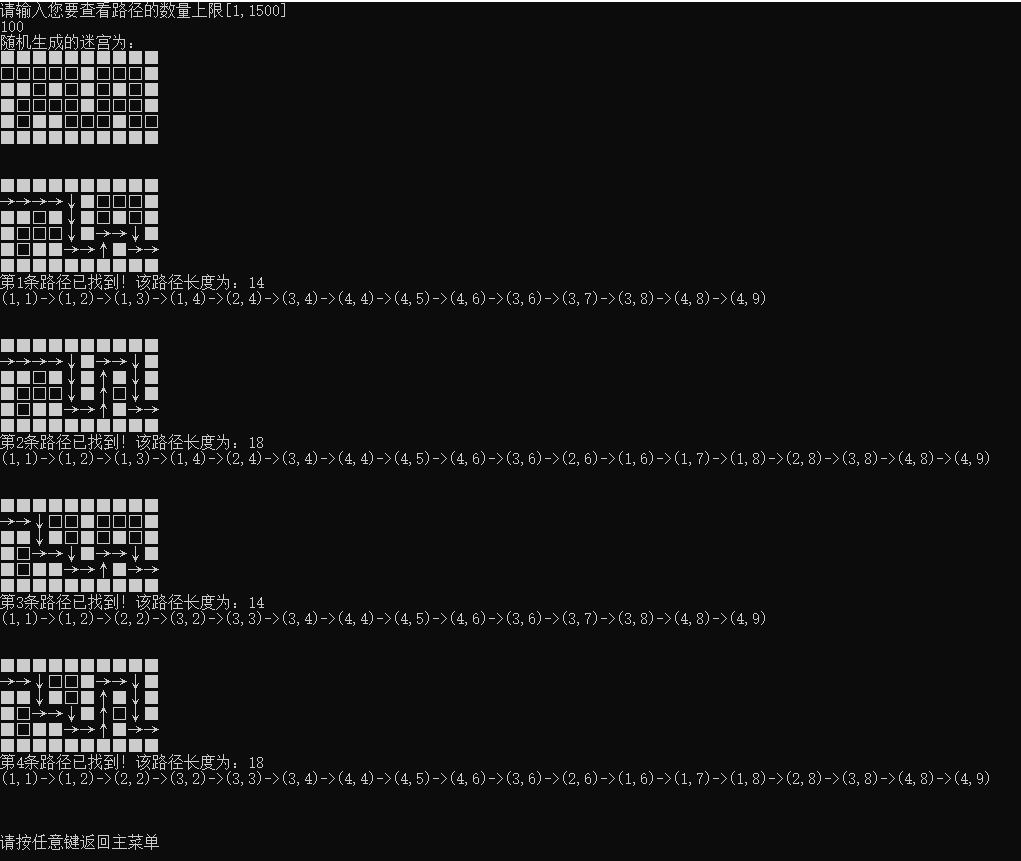
5）动态显示路径



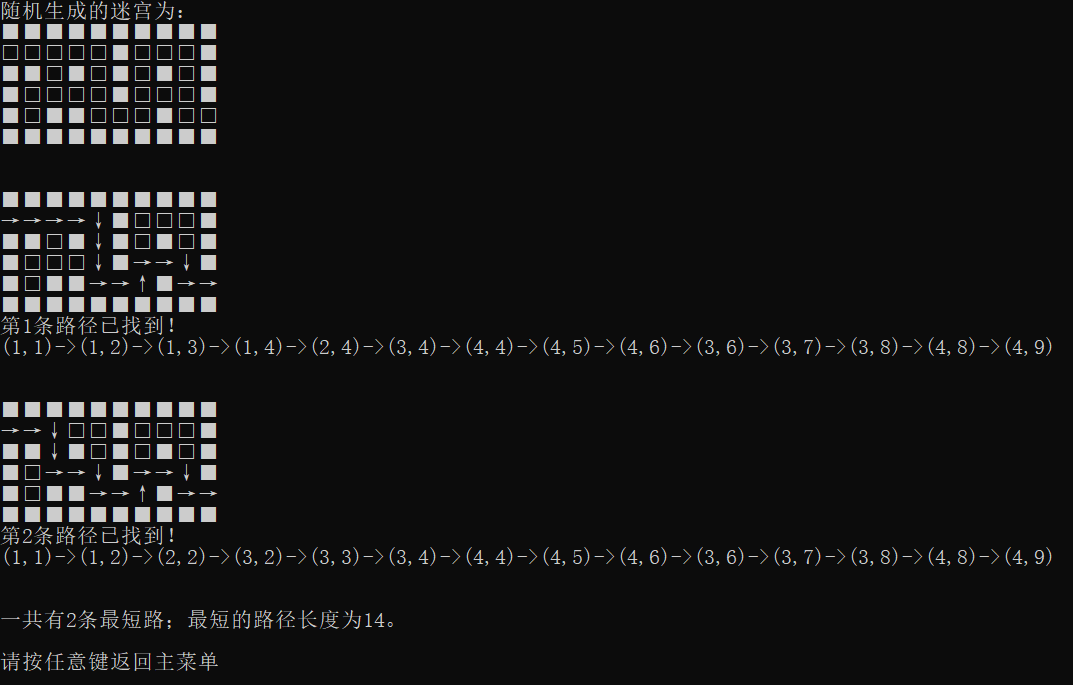
6）动态显示路径



7）输出所有路径

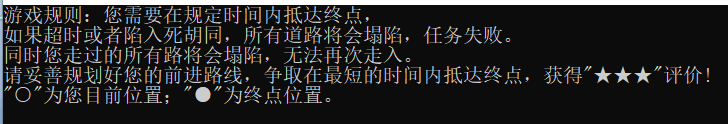


8）输出最短路径

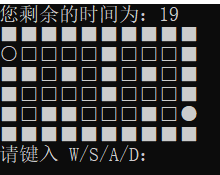


9）手动迷宫游戏

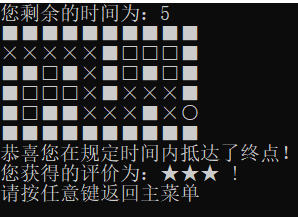
游戏规则



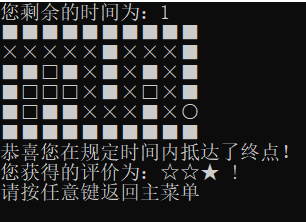
初始化



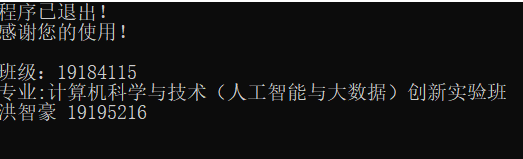
三星评价走法



一星评价走法



10）退出迷宫程序



**五 ．调试分析**

1. 在Random函数中，未清空vis数组 ，会导致程序出现问题
2. 运行完dfs函数并输出所需信息后，要将栈和vis数组以及count清零，以免影响后续功能的实现。
3. DFS回溯的时候，要特判下一个点是不是死路，以免将死路的标记重置导致程序陷入死循环
4. 用户输入时要检测用户输入是否合法
5. 本题中Random，PrintMaze，PrintPaze的时间复杂度为O(M\*N)

即本题的空间复杂度亦为O(M\*N) （栈所占最大空间）

**六 ．测试结果**

**功能点1：动态显示**

输入文件input.txt内的迷宫数据为：

6 10

1 1 1 1 1 1 1 1 1 1

0 0 0 0 0 1 0 0 0 1

1 1 0 1 0 1 0 1 0 1

1 0 0 0 0 1 0 0 0 1

1 0 1 1 0 0 0 1 0 0

1 1 1 1 1 1 1 1 1 1

遍历顺序为 3 2 1 4（左下右上）

期望结果：

■■■■■■■■■■

→→→→↓■□□□■

■■□■↓■□■□■

■□□□↓■→→↓■

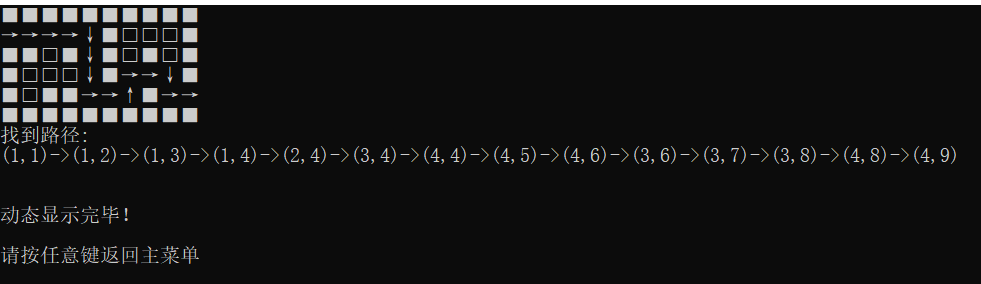
■□■■→→↑■→→

■■■■■■■■■■

路径:

(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(3,4)->(4,4)->(4,5)->(4,6)->(3,6)->(3,7)->(3,8)->(4,8)->(4,9)

实测结果：



经校验，实测结果与期望结果相符

测试通过！

**功能点2：多条路径：**

输入文件input.txt内的迷宫数据为：

7 15

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 0 0 1 0 0 0 1 1 0 0 0 1

1 1 0 1 0 0 0 1 0 1 1 0 1 1 1

1 0 0 0 1 1 0 0 0 1 1 0 0 0 1

1 0 1 0 0 0 0 1 1 0 0 0 1 0 1

1 0 1 0 1 1 0 0 0 0 1 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

遍历顺序：1 2 3 4（右下左上）

期望结果：

■■■■■■■■■■■■■■■

→→→→↓■□□□■■□□□■

■■□■→→↓■□■■□■■■

■□□□■■↓□□■■□□□■

■□■□□□↓■■→→↓■□■

■□■□■■→→→↑■→→→→

■■■■■■■■■■■■■■■

第1条路径已找到！该路径长度为：20

(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(2,5)->(2,6)->(3,6)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(5,11)->(5,12)->(5,13)->(5,14)

■■■■■■■■■■■■■■■

→→→→↓■□□□■■□□□■

■■□■→→↓■□■■□■■■

■□□□■■↓□□■■→→↓■

■□■□□□↓■■→→↑■↓■

■□■□■■→→→↑■□□→→

■■■■■■■■■■■■■■■

第2条路径已找到！该路径长度为：22

(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(2,5)->(2,6)->(3,6)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(3,11)->(3,12)->(3,13)->(4,13)->(5,13)->(5,14)

■■■■■■■■■■■■■■■

→→→→↓■→→↓■■□□□■

■■□■→→↑■↓■■□■■■

■□□□■■↓←←■■□□□■

■□■□□□↓■■→→↓■□■

■□■□■■→→→↑■→→→→

■■■■■■■■■■■■■■■

第3条路径已找到！该路径长度为：26

(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(2,5)->(2,6)->(1,6)->(1,7)->(1,8)->(2,8)->(3,8)->(3,7)->(3,6)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(5,11)->(5,12)->(5,13)->(5,14)

■■■■■■■■■■■■■■■

→→→→↓■→→↓■■□□□■

■■□■→→↑■↓■■□■■■

■□□□■■↓←←■■→→↓■

■□■□□□↓■■→→↑■↓■

■□■□■■→→→↑■□□→→

■■■■■■■■■■■■■■■

第4条路径已找到！该路径长度为：28

(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(2,5)->(2,6)->(1,6)->(1,7)->(1,8)->(2,8)->(3,8)->(3,7)->(3,6)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(3,11)->(3,12)->(3,13)->(4,13)->(5,13)->(5,14)

■■■■■■■■■■■■■■■

→→↓□□■□□□■■□□□■

■■↓■□□□■□■■□■■■

■□→↓■■□□□■■□□□■

■□■→→→↓■■→→↓■□■

■□■□■■→→→↑■→→→→

■■■■■■■■■■■■■■■

第5条路径已找到！该路径长度为：20

(1,1)->(1,2)->(2,2)->(3,2)->(3,3)->(4,3)->(4,4)->(4,5)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(5,11)->(5,12)->(5,13)->(5,14)

■■■■■■■■■■■■■■■

→→↓□□■□□□■■□□□■

■■↓■□□□■□■■□■■■

■□→↓■■□□□■■→→↓■

■□■→→→↓■■→→↑■↓■

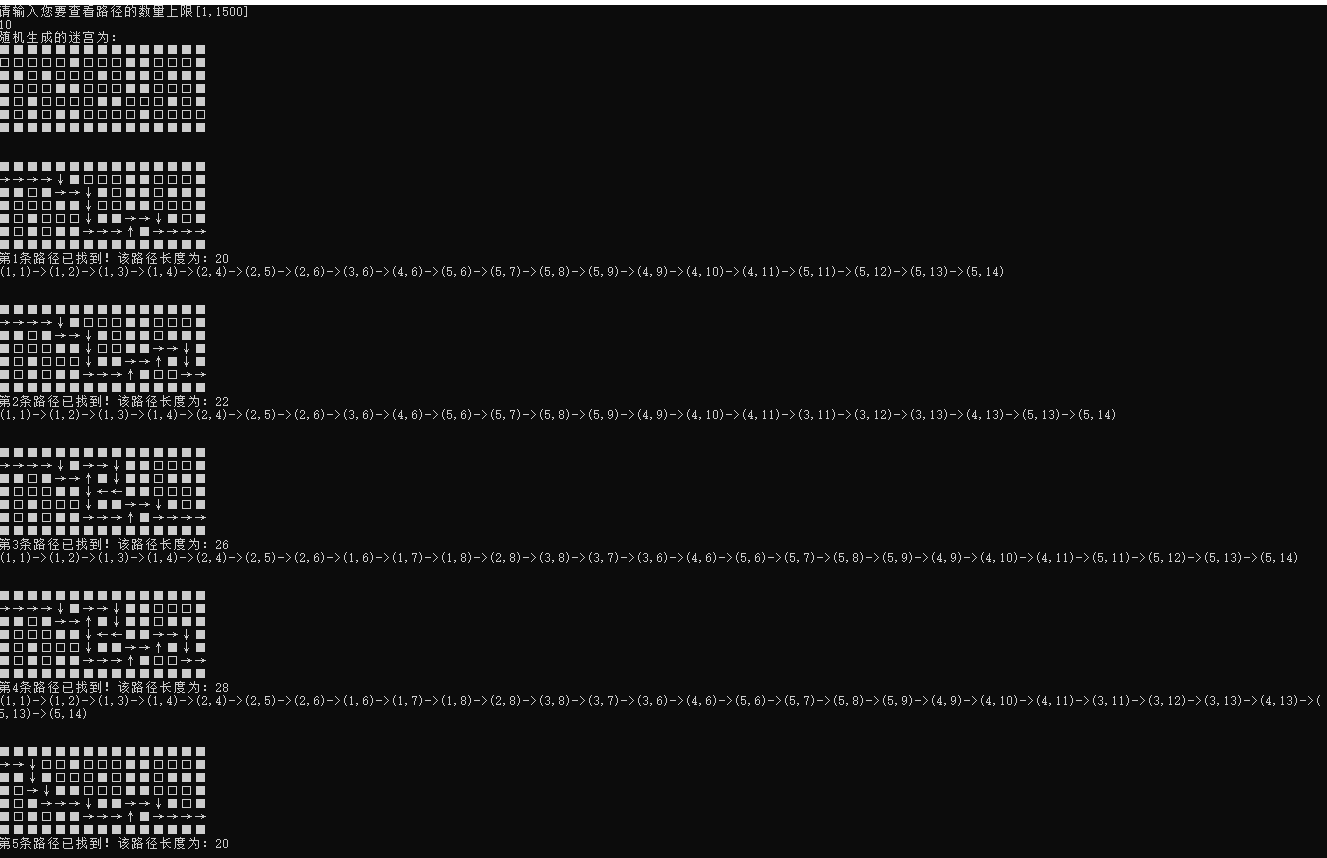
■□■□■■→→→↑■□□→→

■■■■■■■■■■■■■■■

第6条路径已找到！该路径长度为：22

(1,1)->(1,2)->(2,2)->(3,2)->(3,3)->(4,3)->(4,4)->(4,5)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(3,11)->(3,12)->(3,13)->(4,13)->(5,13)->(5,14)

实测结果:



经校验，实测结果与期望结果相符

测试通过！

**功能点3：最短路径**

输入文件input.txt内的迷宫数据为：

7 15

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 0 0 1 0 0 0 1 1 0 0 0 1

1 1 0 1 0 0 0 1 0 1 1 0 1 1 1

1 0 0 0 1 1 0 0 0 1 1 0 0 0 1

1 0 1 0 0 0 0 1 1 0 0 0 1 0 1

1 0 1 0 1 1 0 0 0 0 1 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

期望结果：

■■■■■■■■■■■■■■■

→→→→↓■□□□■■□□□■

■■□■→→↓■□■■□■■■

■□□□■■↓□□■■□□□■

■□■□□□↓■■→→↓■□■

■□■□■■→→→↑■→→→→

■■■■■■■■■■■■■■■

第1条路径

(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(2,5)->(2,6)->(3,6)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(5,11)->(5,12)->(5,13)->(5,14)

■■■■■■■■■■■■■■■

→→↓□□■□□□■■□□□■

■■↓■□□□■□■■□■■■

■□→↓■■□□□■■□□□■

■□■→→→↓■■→→↓■□■

■□■□■■→→→↑■→→→→

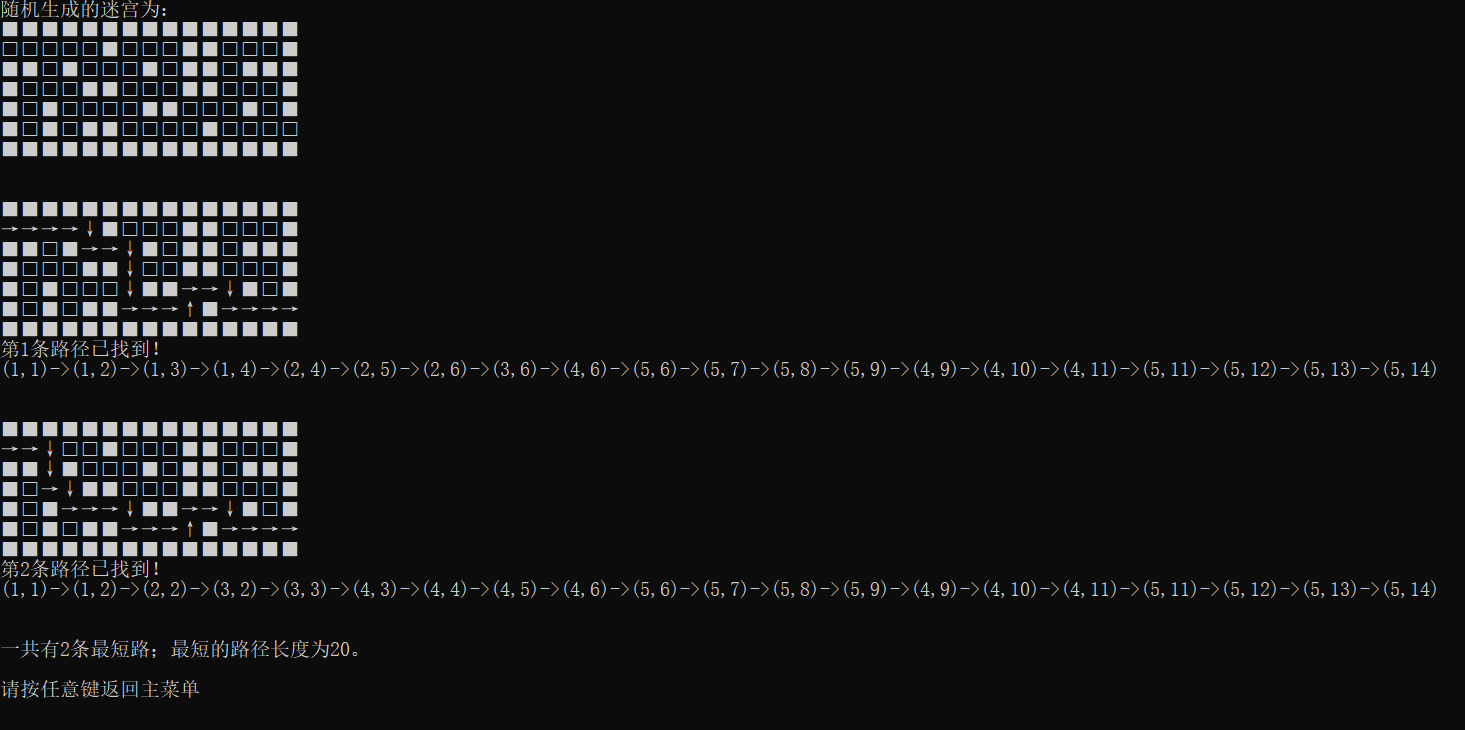
■■■■■■■■■■■■■■■

第2条路径

(1,1)->(1,2)->(2,2)->(3,2)->(3,3)->(4,3)->(4,4)->(4,5)->(4,6)->(5,6)->(5,7)->(5,8)->(5,9)->(4,9)->(4,10)->(4,11)->(5,11)->(5,12)->(5,13)->(5,14)

一共有2条最短路；最短的路径长度为20。

实测结果：



经功能点2中的全部路径一一对比校验，

实测结果与期望结果相符

测试通过！

**功能点4：手动迷宫游戏**

输入文件input.txt内的迷宫数据为：

7 15

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 0 0 1 0 0 0 1 1 0 0 0 1

1 1 0 1 0 0 0 1 0 1 1 0 1 1 1

1 0 0 0 1 1 0 0 0 1 1 0 0 0 1

1 0 1 0 0 0 0 1 1 0 0 0 1 0 1

1 0 1 0 1 1 0 0 0 0 1 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

预期结果：

您剩余的时间为：5

■■■■■■■■■■■■■■■

×××××■□□□■■□□□■

■■□■×××■□■■□■■■

■□□□■■×□□■■□□□■

■□■□□□×■■×××■□■

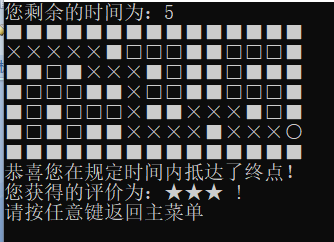
■□■□■■××××■×××○

■■■■■■■■■■■■■■■

恭喜您在规定时间内抵达了终点！

您获得的评价为：★★★ !

实测结果：



经校验，实测结果与期望结果相符

测试通过！

**七 ． 附录**

mazegame.c //主程序

input.txt //输入文件

mazeDynamic\_stack.h //栈文件