**SEP上机考试题目描述**

**题目描述**

小雪人迷路了，空地上有许多对小雪人来说很危险的火堆，需要你找到一条远离火堆的路，帮助小雪人安全回家。空地的大小为M\*N，小雪人初始在左上角（即单元格(0, 0)），小雪人的家在右下角（即单元格(M – 1, N - 1)）。如果单元格的值为0，则表示这个单元格可以通行；如果单元格的值为1，则表示这个单元格上有火堆。小雪人可以往上下左右四个方向移动，但不可以移出边界。求解过程分为两步：

（1）对于每个单元格，定义其 **安全程度** 为：该单元格到所有火堆所在单元格的曼哈顿距离（单元格(x1, y1)和单元格(x2, y2)之间的曼哈顿距离等于 | x1 – x2 | + | y1 – y2 |）的 **最小值**。求出每个单元格的安全程度；

（2）对于一条路径，定义其 **安全程度** 为：路径上所有单元格的安全程度的 **最小值**。找出一条 **从左上角出发、右下角结束且安全程度大于等于给定值** 的路径。

对于多条符合要求的路径，你只需要输出任意一条即可。

火堆的数量，即值为1的单元格的数量至少为1。

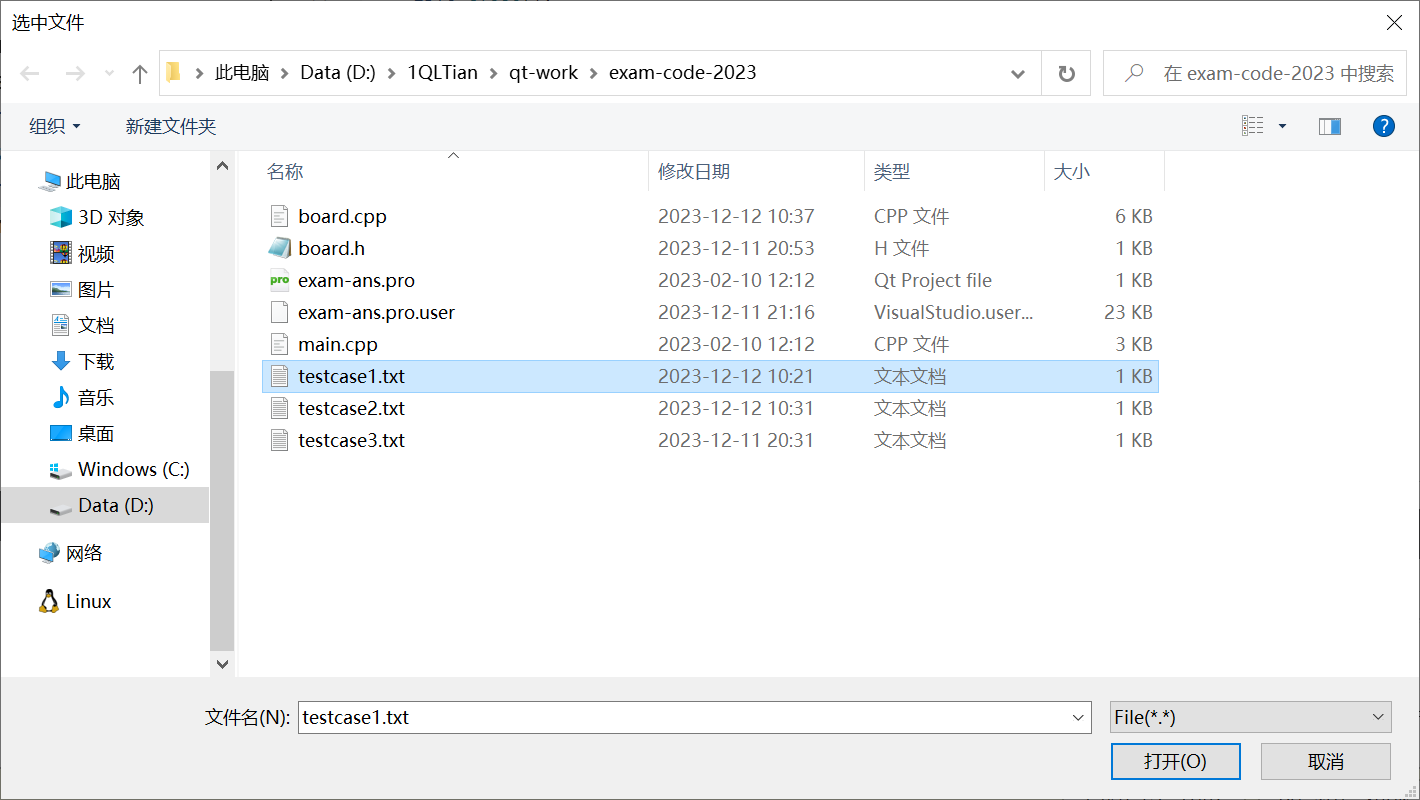
**满分100分。**

**详细功能要求（包含分数细则）**

为你的程序设计图形化界面，其应该具有下列功能：

1. 使用文件浏览器选择读入文件。（10分，现场检查评分）

文件包含火堆分布情况和要求的安全程度值，具体格式在下文（“输入文件格式”）所述。读入文件需要通过按钮或者是QMenuBar来实现。用户通过按钮或者menu bar，选择文件加载输入文件。



输入文件的一个样例为：

3 3

0 0 1

0 0 0

0 0 0

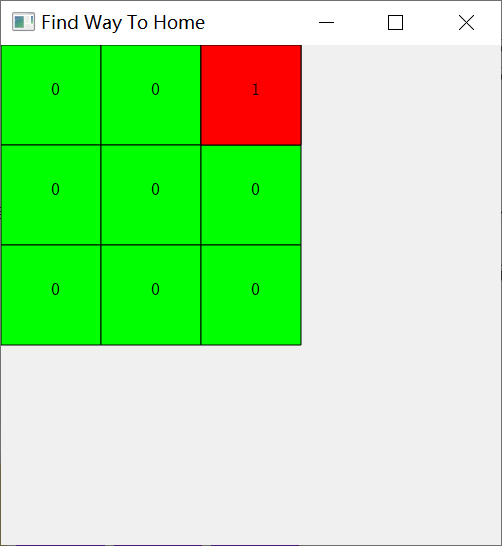
2

表示空地大小为 3x3，且单元格(0, 2)为火堆，要求输出一条安全程度大于等于2的路径。

1. 设计图形化界面展示空地中的火堆分布情况，画出整个矩形空地地图并在其中画出火堆位置，并用颜色标识出火堆 。 （10分，现场检查评分）

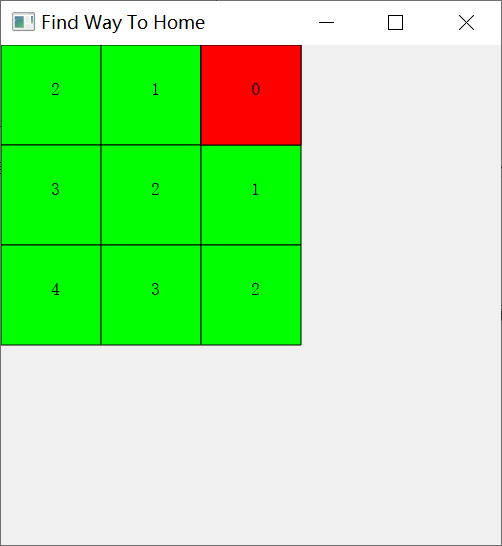
如下图所示，地图上需要显示（至少）两种不同的颜色，一个表示存在火堆的区域，一个表示可以通行的区域。如下图，红色表示火堆区域，绿色表示可以通行的区域。

该阶段可以显示数字，也可以不显示数字。



1. 按下键盘上的T键显示每个单元格的安全程度，将数字显示在每个单元格中。（10分，现场检查评分）

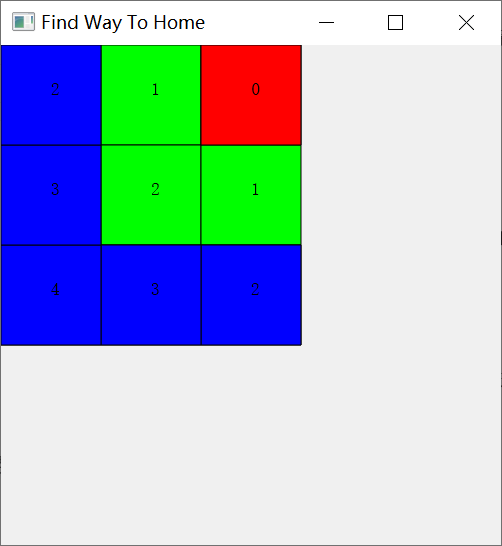
如下图所示，火堆所在单元格的安全程度为0，可通行单元格的安全程度按照之前给出的曼哈顿距离公式计算即可。



1. 按下键盘上的P键开始寻找路径，在你设计的图形化界面中展示你找到的具有特定安全程度的路径。（10分，现场检查评分）

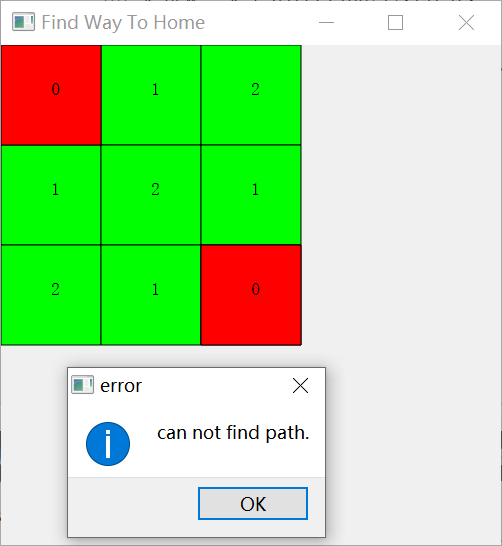
例如，在下例中寻找一条安全程度大于等于2的路径，如下图所示，蓝色的单元格组成的路径表示一个可行解。

（**注意**：这部分，只要能展示路径，就能获得10分）



如果不能找到一条路径，大于等于特定的安全程度，则弹窗显示”can not find path.”。

例如，在下例中寻找一条安全程度大于等于1的路径，而路径的起点和终点所在单元格的安全程度都为0，因此无法找到一条符合要求的路径，弹窗显示，如下图所示。



1. 根据给定样例输出正确结果 。（20分，现场检查评分）

样例输入在题目最后。现场大家需要能够针对样例，在界面上显示所有单元格的安全程度（正确结果），并且在终端上显示一条大于等于给定安全程度值的路径（正确结果），就可以能获得20分。

如果因为前面GUI的不能在界面上展示结果，请确保能够在终端上输出所有正确结果，方便助教线下评分。

终端上输出合法路径的格式为：

a(0, 0) a(0, 1) … a(0, N-1)

...

a(M-1, 0) a(M-1, 1) … a(M-1, N-1)

(x1, y1)

…

(xn, yn)

其中，a(x, y)为单元格(x, y)的安全程度，矩阵输出结束后需要输出一个换行符。

接着按顺序输出一条由若干个单元格坐标组成的符合要求的路径，以(0, 0)开始，以(M – 1, N - 1)结束。

如果不能找到一条符合要求的路径，则输出”can not find path.”。

1. 通过额外的隐藏测试用例 （40分，由助教后台评分）

完成上面的步骤后，将代码上传到Canvas上，由助教后台完成剩余测试用例的评分。

**功能逻辑解释**

由QT实现的GUI部分占据40分（上面的功能要求的1~4点）。

实现正确的功能逻辑占据60分（上面的功能要求的5、6点）。

你的程序应能完成下述功能：

1. 载入输入文件，使用任意（你熟悉的）数据结构存储；
2. 按照单元格的安全程度的计算方式，计算每个单元格的安全程度，并记录下安全程度矩阵；
3. 使用自己设计的算法在给定的空地地图上找到任意一条满足条件的路径，并记录下路径；
4. 输出安全程度矩阵与你找到的路径至标准输出。如果你能保证GUI始终能够正确显示结果的话，这一步可以省略。但是，强烈建议实现这一点，以便助教能够在你的程序GUI出现错误时直接检查标准输出结果。

**Get Started**

文档会随着代码框架一起下发，框架可以帮助你更好地理解题目要求，并且实现更好的代码结构，帮助更快开始着手写代码。

框架内容包括如下部分

* GUI部分:
  + 框架代码仅提供了最外部的窗口部分。同学们需要自行实现剩余的代码。
* 算法实现接口部分：
  + 框架部分提供了对应子问题的函数接口transferMatrix和findPath, 同学们可以在这两个函数中实现程序的算法逻辑，并且将结果返回到上层调用代码进行绘图。

说明：如果代码框架的实现出现问题，请立即现场联系助教。此外，如果你倾向于从头（或者用自己准备的框架）写，同样是可以的。但是尽量对齐代码框架的输出格式要求（这对于助教测试隐藏用例非常重要！）

**输入文件格式**

文件的基本格式如下，一共M+2行（M为地图矩阵的行数）：

M N

a(0, 0) a(0, 1) … a(0, N-1)

...

a(M-1, 0) a(M-1, 1) … a(M-1, N-1)

S

其中，M表示迷宫矩阵的行数，N表示迷宫矩阵的列数，之后M行表示每一行中的N个单元格的数字，数字即代表该单元格的种类，**如果是0则代表该单元格为可通行单元格，是1则代表该单元格为火堆。**最后一行的数字为路径的安全程度S。

**给定的M和N是介于1到15之间的正整数。**

**样例**

下面给出2个示例，2个样例也会同时随着代码框架一同下发，命名为testcase1.txt和testcase2.txt。请根据以下样例开始编写您的代码，助教会当场检查这2个样例的GUI显示结果。

样例1：

5 5

0 0 0 1 1

0 0 0 0 1

0 0 0 0 0

1 0 0 0 0

1 1 0 0 0

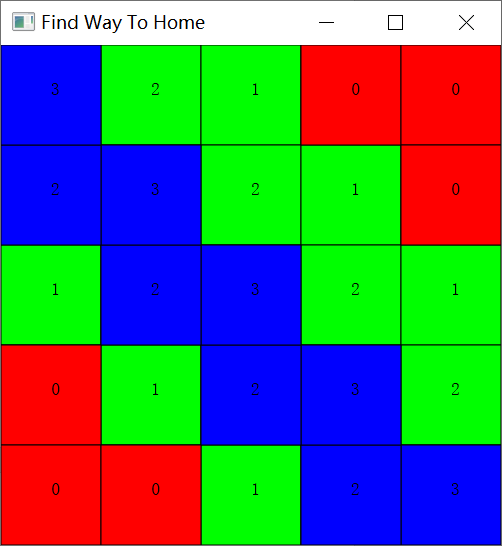
2

测试用例1显示的GUI图像，以及其中一条求解出的路径，应如图所示：

本样例有多条符合要求的路径，展示的只是其中一条，其他符合要求的路径也是对的







对应的终端输出为：

3 2 1 0 0

2 3 2 1 0

1 2 3 2 1

0 1 2 3 2

0 0 1 2 3

(0, 0)

(1, 0)

(1, 1)

(2, 1)

(2, 2)

(3, 2)

(3, 3)

(4, 3)

(4, 4)

样例2：

3 3

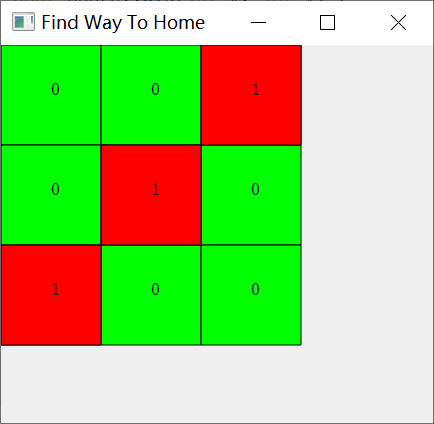
0 0 1

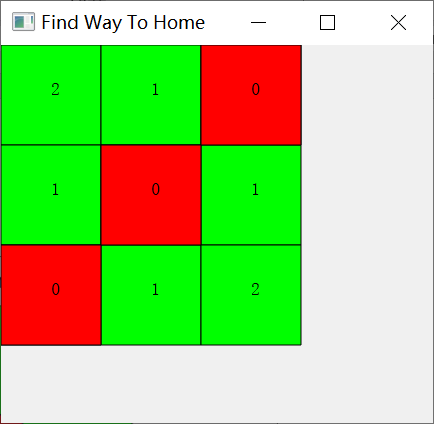
0 1 0

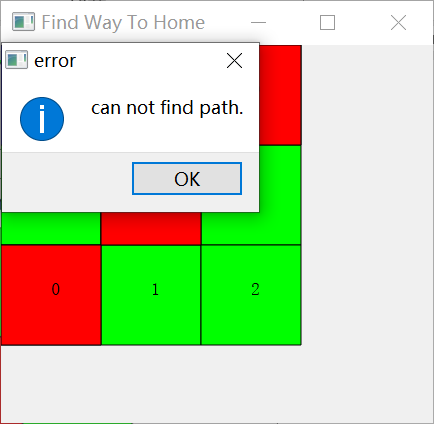
1 0 0

1

测试用例2显示的GUI图像，应如图所示：







对应的终端输出为：

2 1 0

1 0 1

0 1 2

can not find path.