

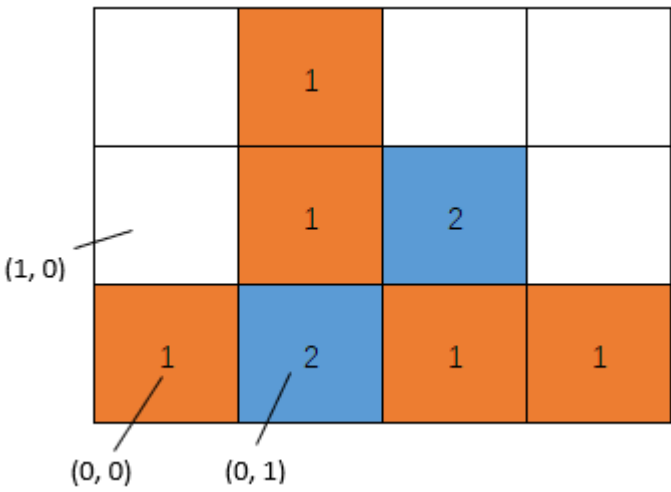
题目描述—开心消消乐

欢迎来到“开心消消乐”的世界！这是一款简单而又充满乐趣的消除类游戏。玩家需要通过交换相邻的元素，使三个或更多相同的元素连成一线，以达成消除效果。每次消除后，上方元素会发生掉落，填补空缺。

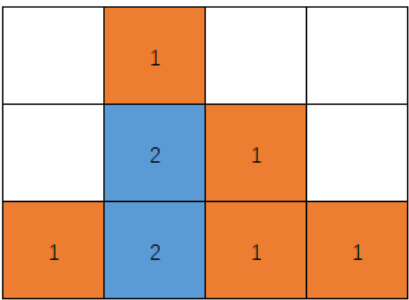
游戏规则（**请仔细阅读，规则可能与普通的消消乐不同**）：

- 1. 操作：交换相邻的两个元素，形成横向或纵向三个或更多相同元素的组合
 - a) 为了简易起见，每次交换**仅能选择两个横向**相邻格子进行交换。可以与空格进行交换（不能选择两个空格进行交换）。如果交换后下方没有方格则会发生掉落（此时先掉落，再判定消除）。交换后即使没有发生消除也可以交换。

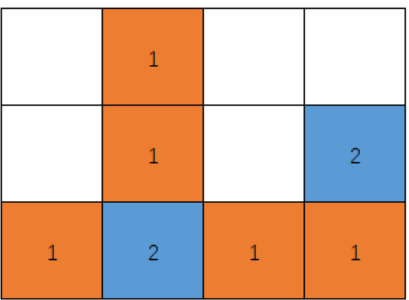
原图（左下为（0，0），第一维为行，第二维为列）：



合法的交换情况 1：（1,1)和(1,2)



合法的交换情况 2：（1,2)和(1,3)



合法的交换情况 3: (2,0)和(2,1)

1			
	1	2	
1	2	1	1

交换:

不合法的交换情况: (0,2)和(1,2)

	1		
	1	1	
1	2	2	1

掉落:

1	1	2	
1	2	1	1

- b) 在没有方块处于掉落状态时，如果在一横行或者竖列上有连续三个或者三个以上相同颜色的方块，则它们将立即被消除。
- c) 消除后方块将会掉落，掉落可能导致新的方块消除，但掉落过程中不会触发消除。掉落导致消除后也可能会触发新的掉落。
- d) 具体消除实例:
- 横向三个和三个以上

1	1	1

1	1	1	1

- 纵向三个和三个以上
- 出现行和列都满足条件且共享某个方块时，行和列都被消除，如

	1		
	1		
1	1	1	1

	1		1
	1		1
1	1	1	1

图中橙色格子都会被消除

- 由于降落导致消除

原图：

	1	1	
2	1	2	2

，交换(0,0)和(0,1)

交换结果：

	1	1	
1	2	2	2

消除 1：

	1	1	
1			

降落：

1	1	1	

消除 2：

2. 计分：每次消除获得 $3+(x-3)*2$ 分，其中 x 是由于一次消除的方块数。注意，一次交换可能导致多次消除，例子如下：

1	1	1

一次消除，计 3 分

1	1	1	1

一次消除，计 5 分

	1		
	1		
1	1	1	1

一次消除，计 9 分

	1		2
	1		2
3	1	3	2

两次消除（颜色不同且不相连），计 3+3=6 分

	1		1
	1		1
2	1	2	1

两次消除（不相连），计 3+3=6 分

	1	1	
1	2	2	2

两次消除，计 3+3=6 分

下面您需要开发一款开心消消乐程序，且具有如下功能：

1. 读取并展示消消乐界面
2. 根据给定的步骤展示消除过程并计算得分

Get Started

压缩包里提供了相应的代码框架，请您基于该框架实现

注意：

1. 您**必须**基于代码框架进行实现，否则可能无法通过后台测试用例
2. 实现过程中请**不要**添加额外文件
3. 在后台测试中，main.cpp、test.cpp、test.h 会被替换，因此，**请不要在这三个文件中实现自己的逻辑**。
4. CMakeLists.txt 在后台也会被替换，若您需要修改 CMakeLists.txt，可以在上传的代码中添加一个 **readme.txt** 进行说明。当后台测试出现问题时，助教将会查看该 readme.txt，并采用您的 CMakeLists.txt 重新进行编译和测试。若您还有其他可能导致后台测试失败的原因（例如采用的 QT 版本为 qt5 等），也请在 readme.txt 中进行备注。

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。
- 算法实现接口部分：
 - State 类：用以存储消消乐状态，包括变量 n 表示棋盘行列数，left 表示剩余步数，board 表示当前的棋盘状态，steps 表示剩余具体步骤。**您需要实现其中的 applySteps 接口和 readFromFile 接口以供测试。**
 - Test 类：实现了基本的测试功能，可以用于检查待测试接口是否实现正确。在您完成实现过后，可以在 main 函数中调用相应的接口，对输出进行检查。**请务必使用该类进行至少一个样例的检查，以避免由于输出格式不对导致扣分！**

说明：如果代码框架的环境出现问题，请立即现场联系助教。

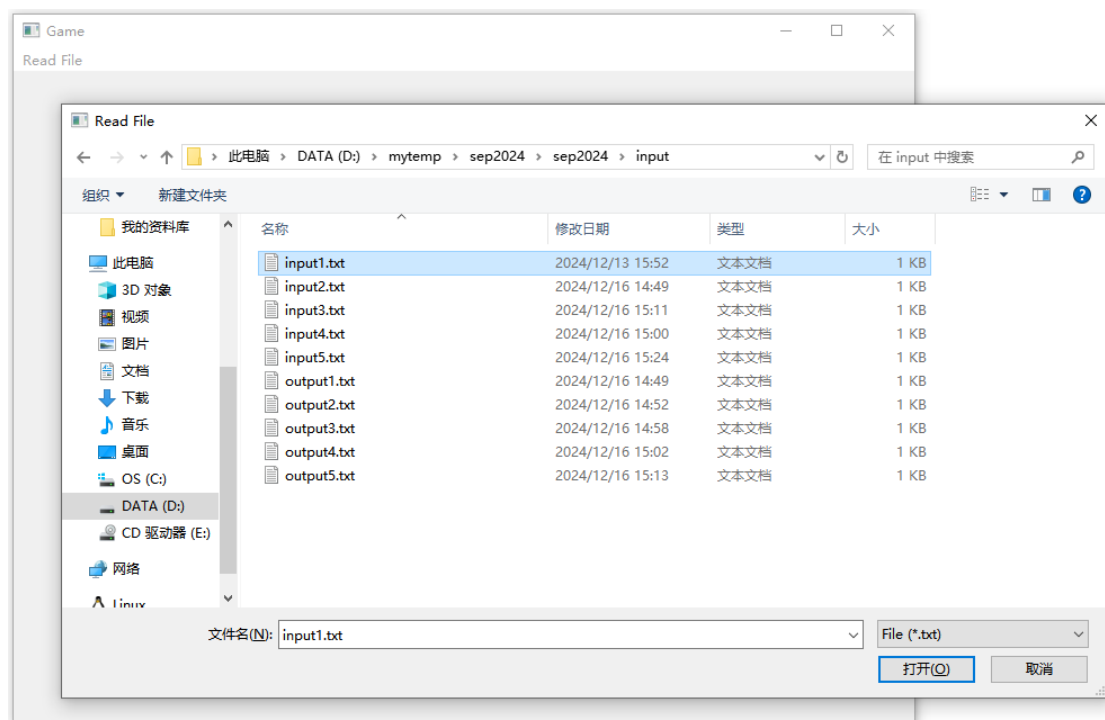
评分标准

请严格按照该标准执行，避免因少实现功能而扣分

现场演示部分（50 分）

为你的程序设计图形化界面，其需要具有下列功能：

1. 使用文件浏览器选择读入文件。
文件包含当前的消消乐情况，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：

```
5 3
00000
00420
00134
00124
11241
03
02
22
```

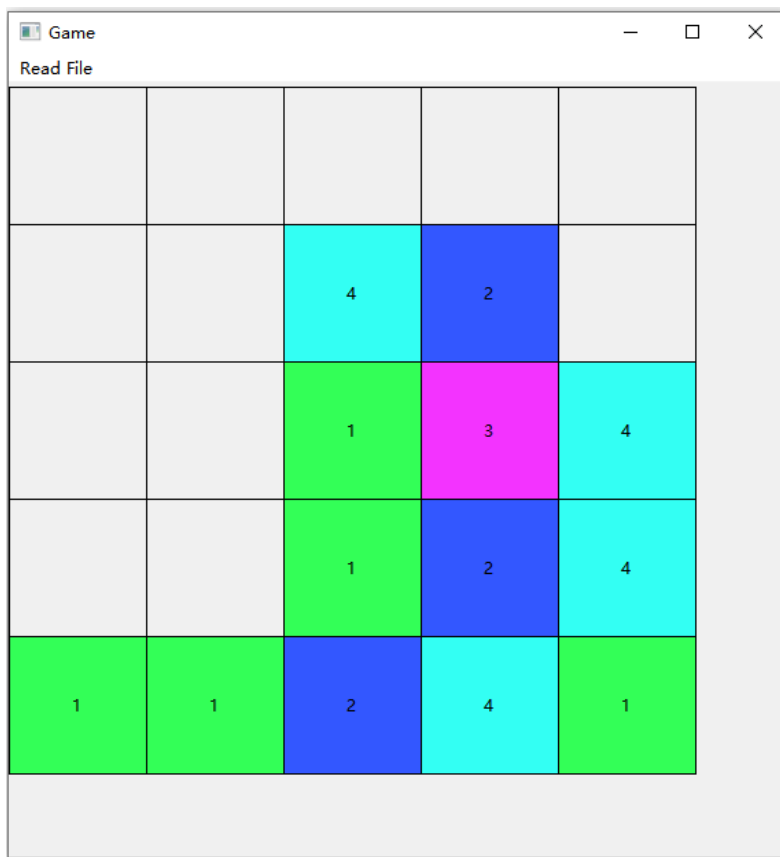
表示第一行输入 N 和 S，N 表示场地大小，例如样例中为 5，表示场地为 5x5，S 表示剩余步数，样例中为 3 表示剩余 3 步。

接下来的方阵中每个数字表示一个不同方块，0 表示空格。

之后的 3 行给定的移动步骤，分别为(0,3), (0,2), (2,2)，其中(0,3)表示(0,3)和(0,4)交换，(0,2)表示(0,2)和(0,3)交换，(2,2)表示(2,2)和(2,3)交换。

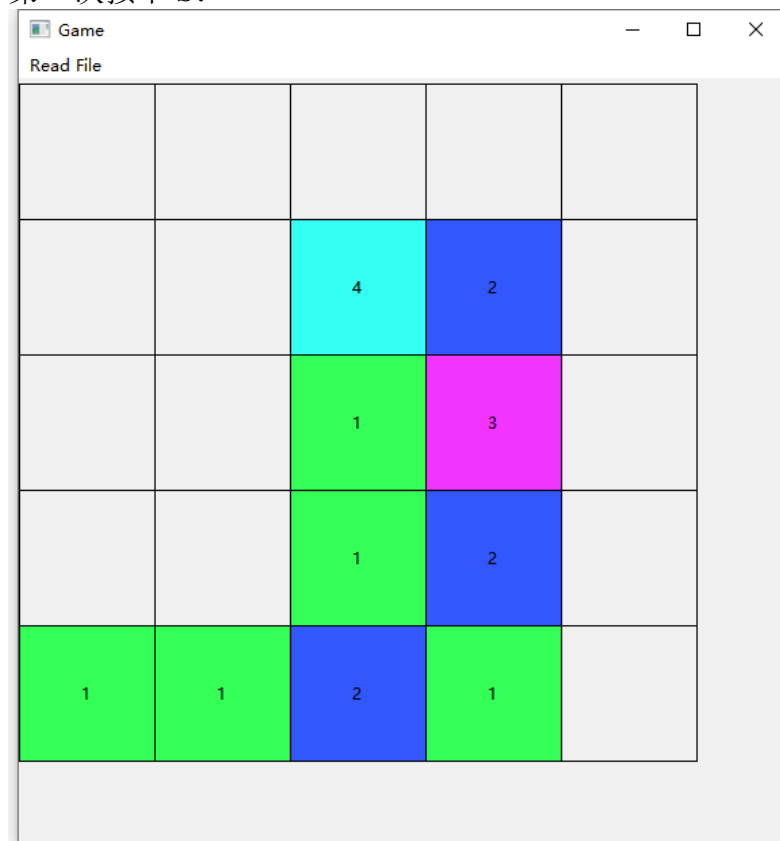
2. 设计图形化界面展示消消乐界面。

如下图所示，不同颜色表示不同的方格，空格不填充颜色，需要显示数字。

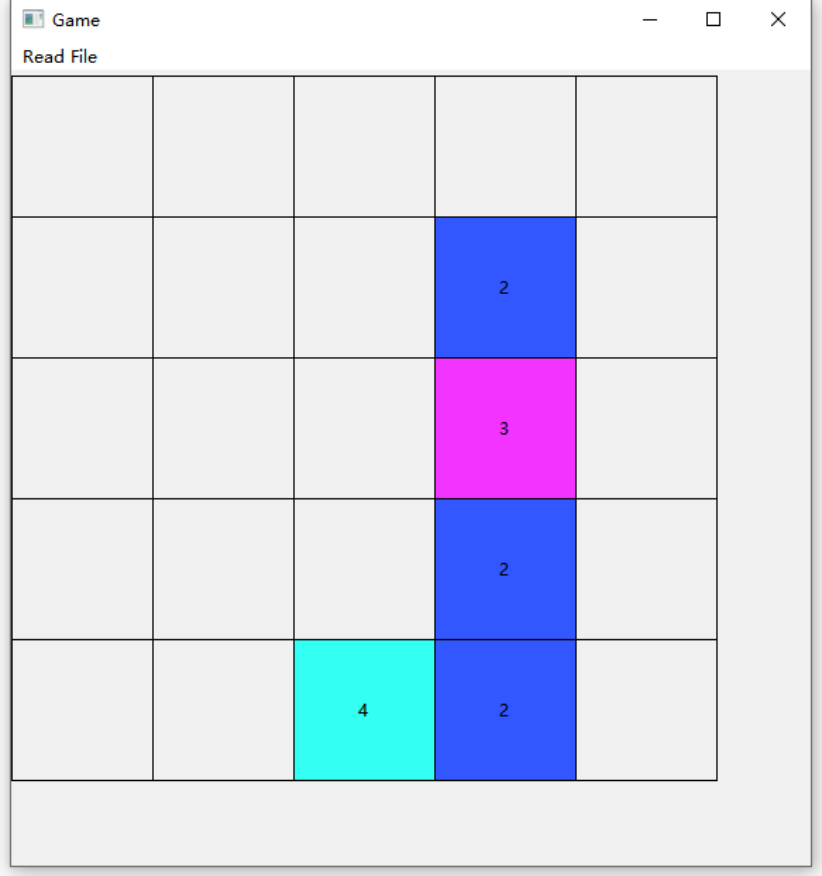


- 按下 S 键展示消除过程，每按一次根据给定步骤交换一次，只需展示交换过后消除和降落完成后的结果。

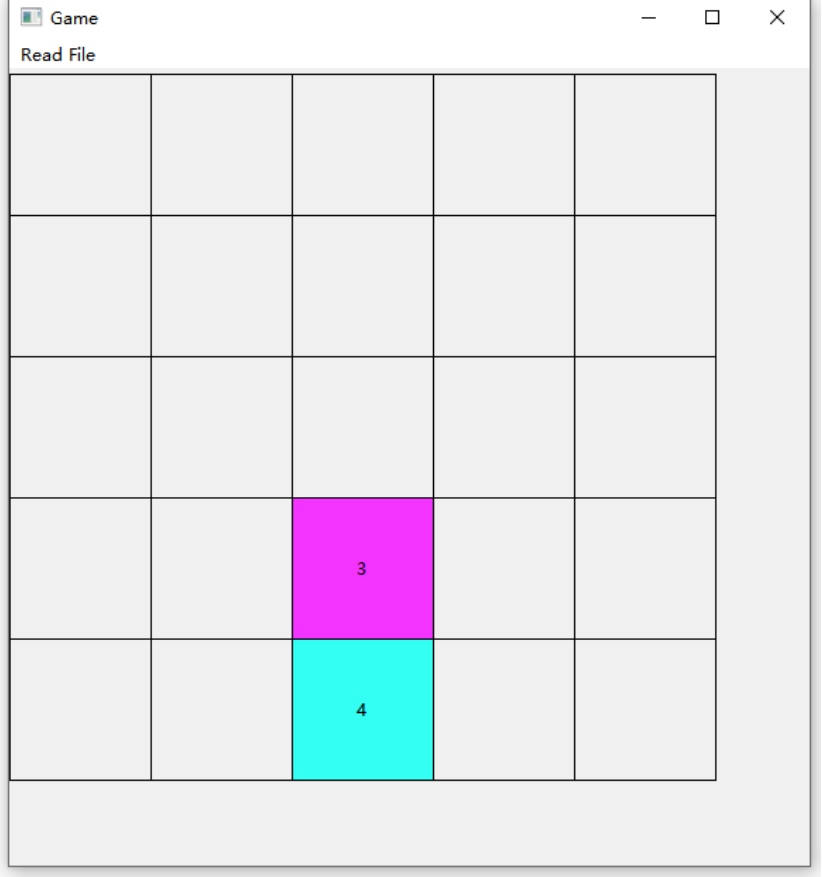
第一次按下 S:



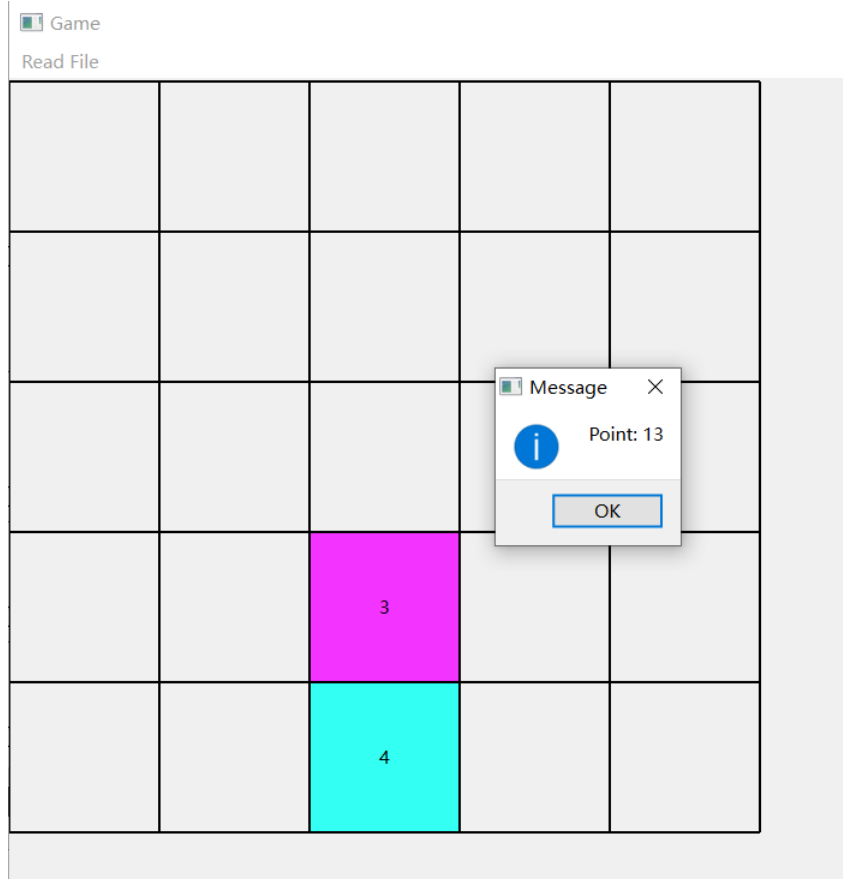
第二次按下 S:



第三次按下 S:



4. 按下键盘上的 R 键根据当前得分弹出窗口。



注意：

1. 演示文件即为样例 1 (若实现未完成，可以演示样例 3，得分点 6 和 8 可各获得 1 分，其余得分点仍能获得全部分数)。
 2. 现场演示流程：
 - a) 按要求打开文件
 - i. 得分点 1：文件打开方式符合要求（10 分）
 - b) 展示图形化界面
 - i. 得分点 2：绘制出方格线（5 分）
 - ii. 得分点 3：不同方格采用了不同颜色，为 0 的方格无颜色（5 分）
 - iii. 得分点 4：正确展示数字（5 分）
 - c) 反复按下 S 键检查步骤
 - i. 得分点 5：每步交换后图像有变化（10 分）
 - ii. 得分点 6：每步交换符合结果（5 分）
 - d) 按下 R 键查看有无弹窗，检查得分是否正确
 - i. 得分点 7：有弹窗（5 分）
 - ii. 得分点 8：弹窗展示的分数正确（5 分）
- 现场演示会按上述流程进行且不会有额外的步骤，除上述得分点外不会有额外的得分点和扣分点

助教后台评分部分

请实现以下函数，助教会对该函数进行后台测试（由助教后台评分）

1. State 类的 applySteps 函数

```
int applySteps();
```

该函数会在 `state` 中应用所有的移动步骤，并处理完成消除和掉落，返回所有移动的得分，并更新 `board` 成员。例如：当 `input` 为

```
5 3
0 0 0 0 0
0 0 4 2 0
0 0 1 3 4
0 0 1 2 4
1 1 2 4 1
0 3
0 2
2 2
```

时，调用该接口会移动以上所有 3 步，并返回得分 13，并更新 `state` 类 `board` 为

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 3 0 0
0 0 4 0 0
```

后台评分标准：

得分点 9: `applySteps` 函数调用后 `state` 类的 `board` 更新正确（25 分）

得分点 10: `applySteps` 函数返回的得分正确（25 分）

输入文件格式

文件的基本格式如下，一共 $N+2$ 行（ N 为地图矩阵的行数）：

```
N S
a(N-1, 0) a(N-1, 1) ... a(N-1, N-1)
...
a(0, 0) a(0, 1) ... a(0, N-1)
x1 y1
x2 y2
... ...
xS yS
```

其中， N 表示行数。

之后 N 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 则代表该单元格为空格，其他则表示对应的方格。

S 为剩余的操作步数。

之后 S 行每行表示一个移动步骤

数据范围：

对于所有用例， $3 \leq N \leq 6$ ， $S \leq 5$ ，颜色种类 $2 \leq k \leq 9$

您的程序应该在 1s 内完成后台测试

部分测试用例中存在如下特殊情况：

部分用例 1: $S=1$

部分用例 2: 仅在底部一排存在方格

部分用例 3: 只会出现 3 个方块消除的情况，不会出现多于 3 个，也不会出现行和列都满足条件共享的情况。

所有用例均不会出现初始状态就存在可以消除的方块的情况

实现提示:

对于后台测试部分，您可以考虑按以下步骤实现您的程序，以便于在没有完成所有情况时获得部分分：

1. 优先仅考虑底部一排存在方格且不会出现多余 3 个方格消除的情况
2. 再考虑添加掉落逻辑，处理所有行均可能存在方格的情况（提示：掉落可能引起新的消除和掉落）
3. 最后再考虑可能出现多于 3 个方格消除的逻辑（提示：遍历，递归）

对于现场演示部分，您可以优先实现界面，待后续函数实现完成后再对接

样例输出文件格式

实现过程中不需要输出到文件，样例的输出文件仅用于您检查您的输出是否正确。文件格式如下：

P N

$a(N-1, 0) a(N-1, 1) \dots a(N-1, N-1)$

...

$a(0, 0) a(0, 1) \dots a(0, N-1)$

其中，P 表示按给定步骤移动的最大得分，N 表示地图矩阵大小，接下来 N 行为按给定步骤移动完成的矩阵。

样例

共 5 个样例，位于下发的 input 文件夹。助教会现场检查样例 1 的 GUI 显示结果。

样例 1:

```
5 3
00000
00420
00134
00124
11241
03
02
22
```

图形界面展示结果见前面题目要求。

样例 1 output 文件如下：

13 5
00000
00000
00000
00300
00400

其中第一行 13 表示按步骤移动的得分，然后是剩余棋盘状态

此外还有 4 个样例文件，具体可见文件：

样例 2：s 为 1 的情况

移动完结果如下：

2		3	
1	1	2	
1	2	2	

样例 3：仅在底部一排存在方格

移动完结果如下：

样例 4：仅会出现 3 个方格消除的情况

移动完结果如下：

		3	
5	3	3	
1	5	5	

样例 5:

移动完结果如下:

					4
			2	4	2
			3	2	3
3	5		4	3	5
2	5	5	1	4	2