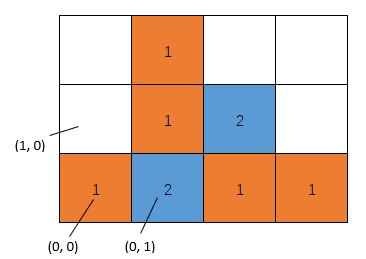
**题目描述—开心消消乐**

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。

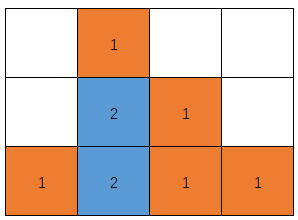
游戏规则（**请仔细阅读，规则可能与普通的消消乐不同**）：

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
   1. 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。

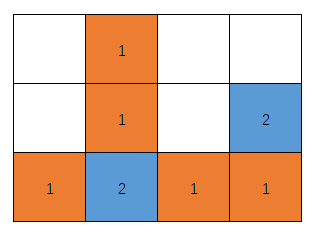
原图（**左下为（0，0），第一维为行，第二维为列**）：



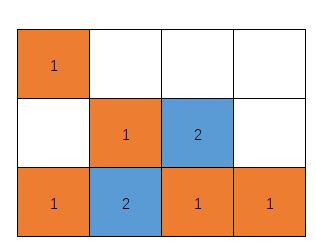
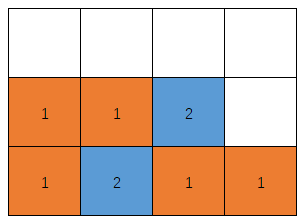
合法的交换情况1：(1,1)和(1,2)



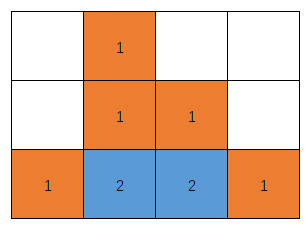
合法的交换情况2：(1,2)和(1,3)



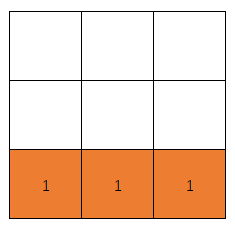
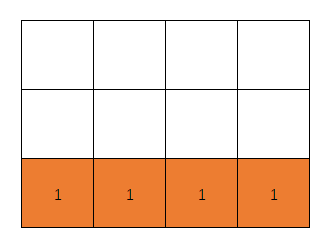
合法的交换情况3：(2,0)和(2,1)

交换： 掉落：

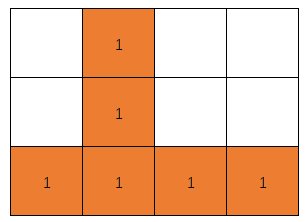
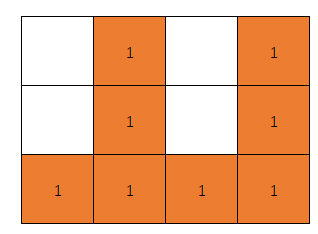
不合法的交换情况：(0,2)和(1,2)



* 1. 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。
  2. 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。
  3. 具体消除实例：
     1. 横向三个和三个以上

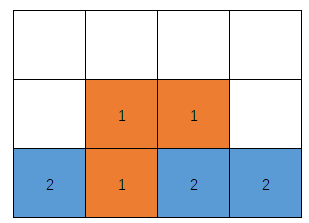
 

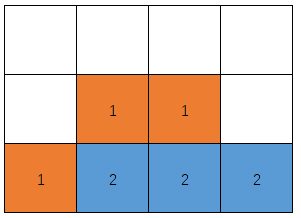
* + 1. 纵向三个和三个以上
    2. 出现行和列都满足条件且共享某个方块时，行和列都被消除，如

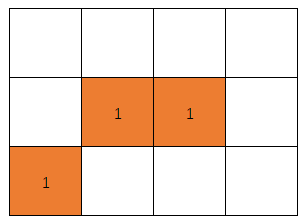
 

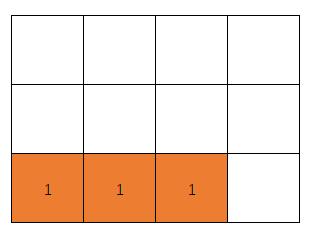
图中橙色格子都会被消除

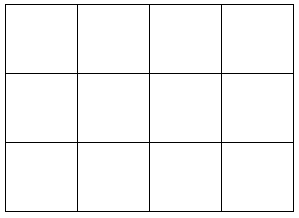
* + 1. 由于降落导致消除

原图： ，交换(0,0)和(0,1)

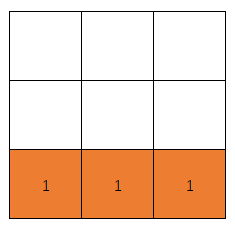
交换结果：

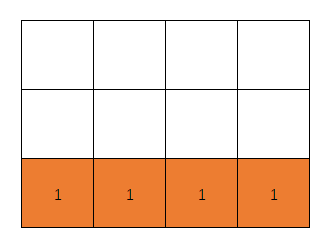
消除1： 

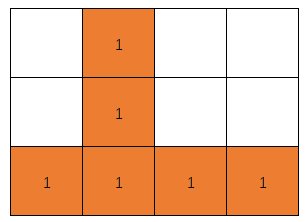
降落： 

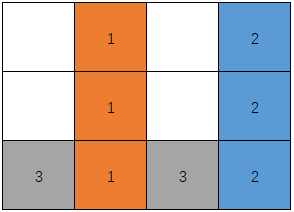
消除2： 

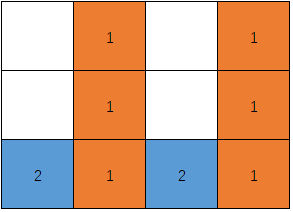
1. 计分：每次消除获得3+(x-3)\*2分，其中x是由于一次消除的方块数。注意，一次交换可能导致多次消除，例子如下：

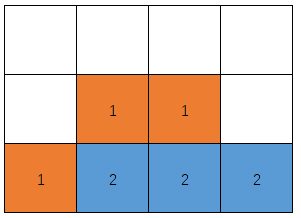
一次消除，计3分

一次消除，计5分

一次消除，计9分

两次消除（颜色不同且不相连），计3+3=6分

两次消除（不相连），计3+3=6分

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**

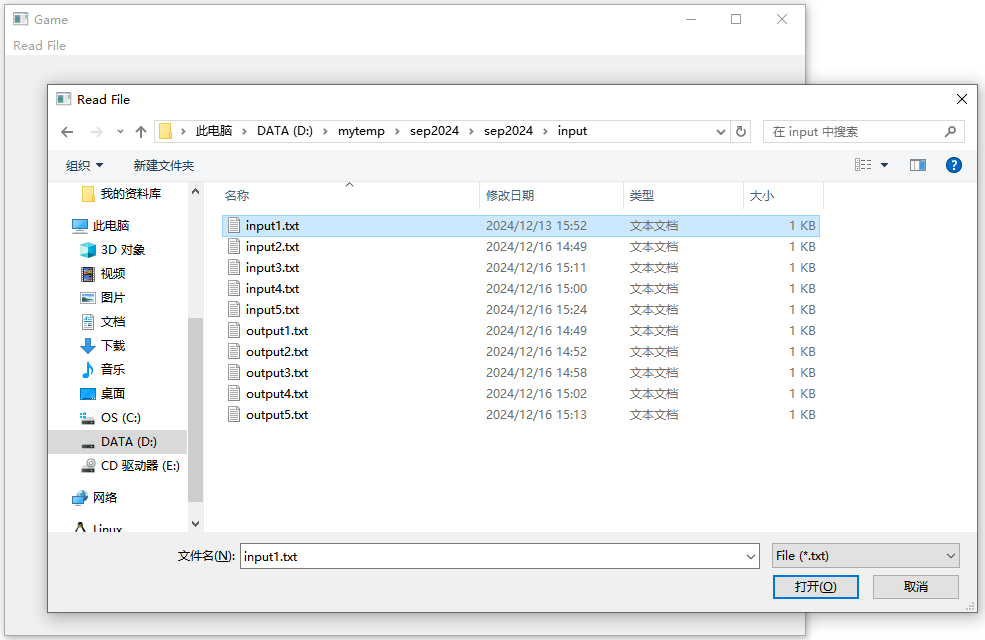
请严格按照该标准执行，避免因少实现功能而扣分

**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

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

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，

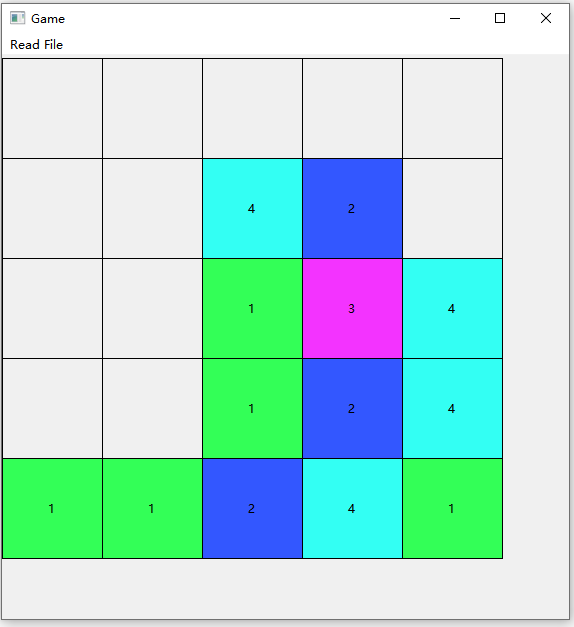
S表示剩余步数，样例中为3表示剩余3步。

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。

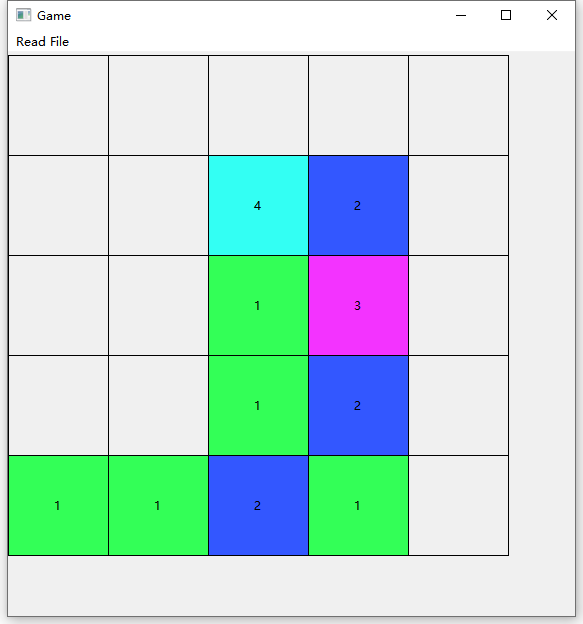
1. 设计图形化界面展示消消乐界面。

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。

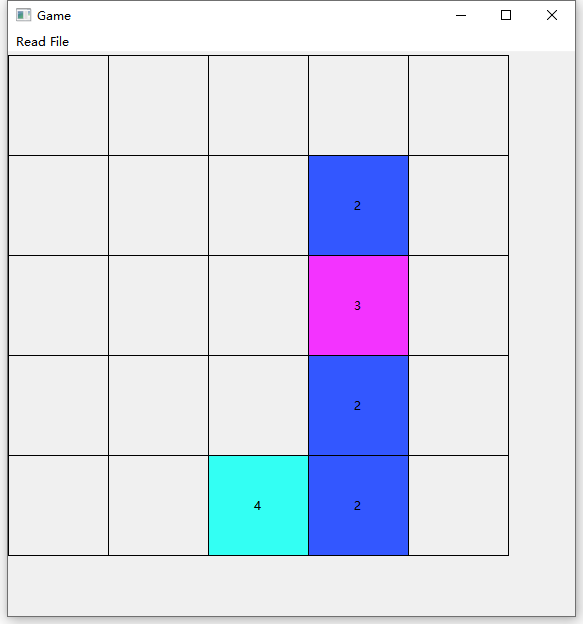


1. 按下S键展示消除过程，每按一次根据给定步骤交换一次，只需展示交换过后消除和降落完成后的结果。

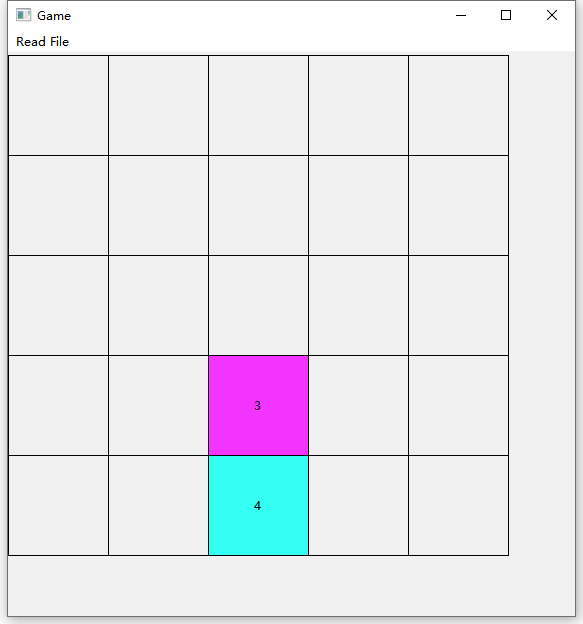
第一次按下S：



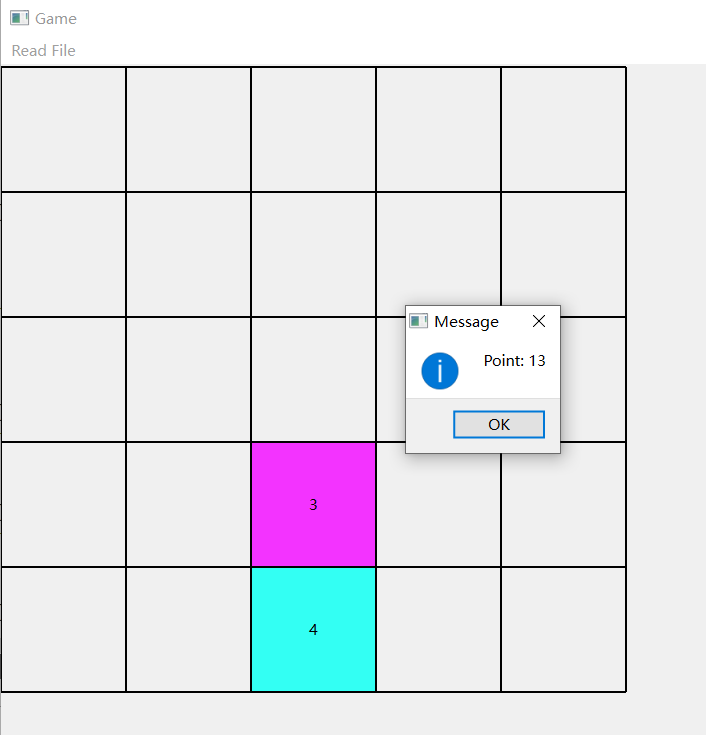
第二次按下S：



第三次按下S：



1. 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：
   1. 按要求打开文件
      1. 得分点1：文件打开方式符合要求（10分）
   2. 展示图形化界面
      1. 得分点2：绘制出方格线（5分）
      2. 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）
      3. 得分点4：正确展示数字（5分）
   3. 反复按下S键检查步骤
      1. 得分点5：每步交换后图像有变化（10分）
      2. 得分点6：每步交换符合结果（5分）
   4. 按下R键查看有无弹窗，检查得分是否正确
      1. 得分点7：有弹窗（5分）
      2. 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<=N<=6，S<=5，颜色种类2<=k<=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) … a(N-1, N-1)

...

a(0, 0) a(0, 1) … 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

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

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

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 3 0 0

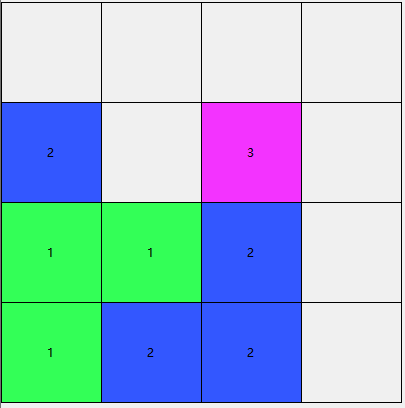
0 0 4 0 0

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：

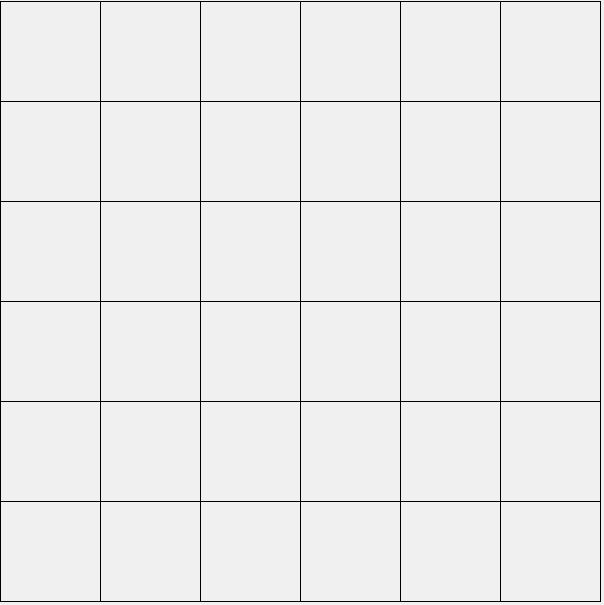
样例2：s为1的情况

移动完结果如下：



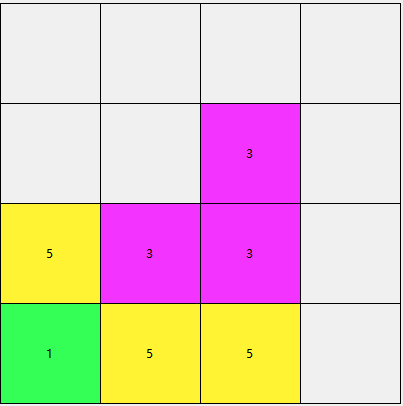
样例3：仅在底部一排存在方格

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：



样例5：

移动完结果如下：

