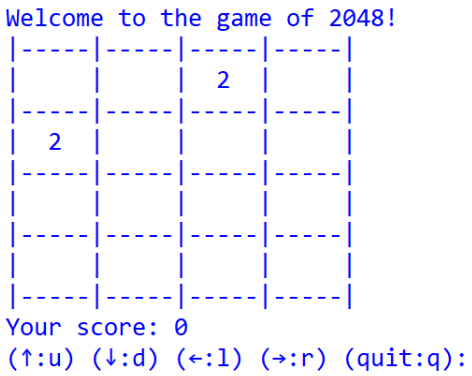
2048 小游戏

游戏的初始状态：生成 4×4 的网格布局（如下图所示），在网格上随机放置两个数字 2，提示用户进行操作（上下左右移动或者放弃游戏）。



移动操作会把网格上现有的数字整体移动到碰墙（边界），同时根据移动方向，判断同一行/列上是否有 2 个连续的数相同：相同则合并成一个双倍的数，且占据 2 个数中比较靠墙的网格位置，累加得分（合并之后的数），合并位置之后的其它数向移动方向填充网格位置；不同则保持不变。每移动一次，游戏会在网格的空白位置随机放置一个数字 2 或 4（概率大约是 4:1）。注意：1、同一行上可能出现 2 组两两相同可以合并的数字；2、同一行上 3 个数字相同，仅合并 2 个靠墙的数字。当网格已填满且无法合并，游戏结束。

游戏结束时，将本次游戏的相关信息汇总到当前文件夹下的表格文件 game2048.csv 中：游戏的开始时间、本次游戏所花费的时间(单位：秒)、游戏得分。提示：csv 文件也是文本文件，每行的各个数据之间以逗号字符分隔。

设计程序的逻辑流程

根据游戏说明，先以伪代码形式设计出程序的整体逻辑流程。这一步很关键，请同学们完成后以注释的形式附加在定义主函数的源程序文件头部。

设计类和类的成员

根据程序的逻辑流程，需定义以下二个类：

1. 网格类 Board

   数据成员：保存网格的数据结构，游戏得分

   成员函数：

   1) 构造函数
   2) 获取得分的访问器函数
   3) 输出网格
   4) 在空白网格位置随机放置一个 2 或 4(概率大约是 4:1)
   5) 在某方向上整体移动数字
   6) 在某方向上合并数字并计分
   7) 判断网格是否已经填满
   8) 判断已填满的网格是否还有可以合并的数字

2. 游戏类 Game2048

   数据成员：网格类对象 board，游戏的开始和结束时间

   成员函数：

   1) 构造函数（游戏开始）

2) 析构函数（游戏结束）
3) 询问用户移动方向：确保用户输入有效字符（假设用户仅输入单个字符，空白符除外）
4) 获取游戏得分
5) 整体移动且合并数字，同时计分
6) 在空白网格位置随机放置一个 2 或 4
7) 输出网格

其中，成员函数 4~7 通过调用数据成员 board 的成员函数完成。

可根据需要，在以上两个类中自行添加数据成员和成员函数。

有关时间的函数

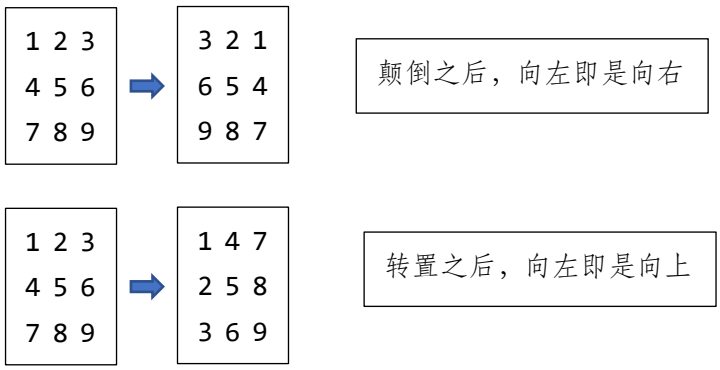
利用 ctime 头文件中声明的函数 time(…)、ctime(…)和 difftime(…)获取游戏开始时间（含日期）和游戏所花费的时间。函数的具体说明可参照：https://blog.csdn.net/qq_38210354/article/details/106910748

以追加方式写入文件

创建文件对象时，指定读写模式为 ofstream::app，例如：ofstream file("game2048.csv", ofstream::app)。

网格数字整体移动与相同数字合并的策略

仅需实现一个方向的移动与合并，其它方向通过适当转换，均可变成该方向的移动与合并。例如：已实现向左移动与合并，向右移动与合并可以先颠倒每行数字，然后调用向左移动与合并函数，最后再把每行数字颠倒回来即可。



提供的附件：

1) Game2048.exe 提供了游戏的可执行程序，通过实际体验获取游戏的实现细节。游戏结束后，也会自动产生文件 game2048.csv 或在该文件中自动添加游戏信息；
2) Game2048.png 提供了游戏运行的示例截图；
3) Game2048_csv.png 提供了 csv 文件的示例截图。