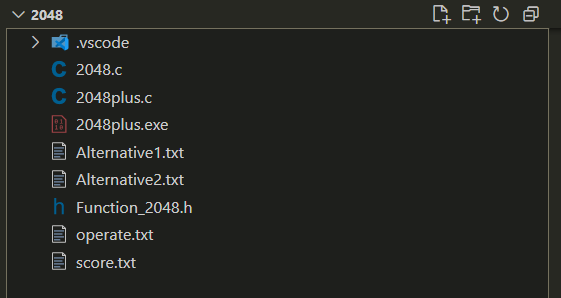
# 2048plus开发文档

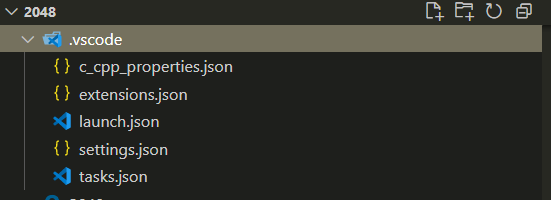
1：项目文件夹介绍

【1】整体文件



【2】.vscode

包含了在vscode中的一些配置



【3】2048.c

期中project，作为参考。

【4】2048plus.c

期末project，源代码中有详细注释。

【5】2048plus.exe

2048plus.c的可执行文件，需要与Alternative1.txt文件在同一个文件夹里方可正常运行。

【6】Alternative1.txt(Alternative2.txt)

包含了要替换的字符序列，可在2048plus.c源文件中通过更改宏定义来改变所要使用的文件。也可直接更改文档中的内容进行自定义。

【7】Function\_2048.h

自定义函数库，源代码中有详细注释。

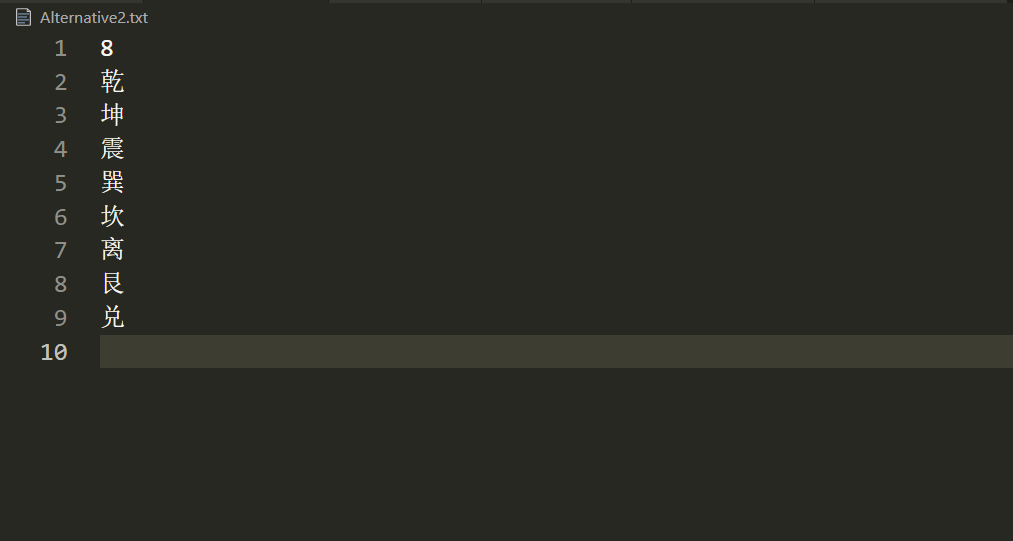
【8】operate.txt和score.txt

分别保存操作和分数的文档，初始为空。

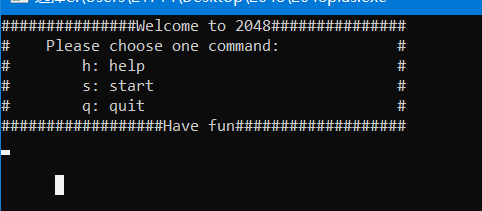
注：文件打开和保存模式都为GBK。

2：游戏运行测试

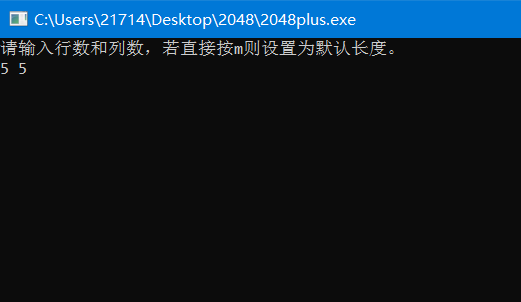
此次测试采用Alternative2.txt



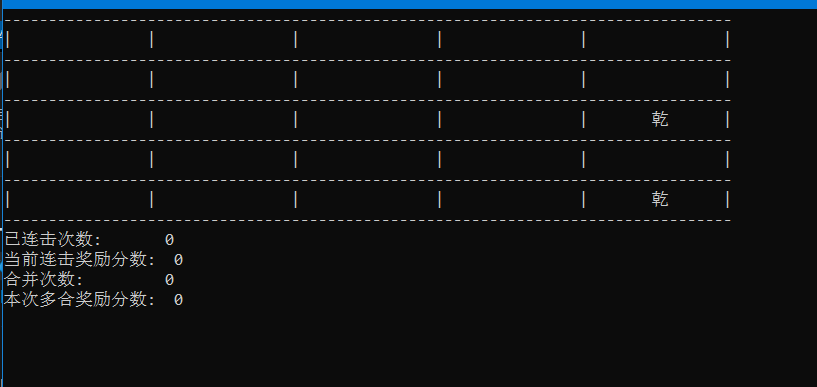
开始界面：



输入s之后进入游戏界面，提示输入行列数。

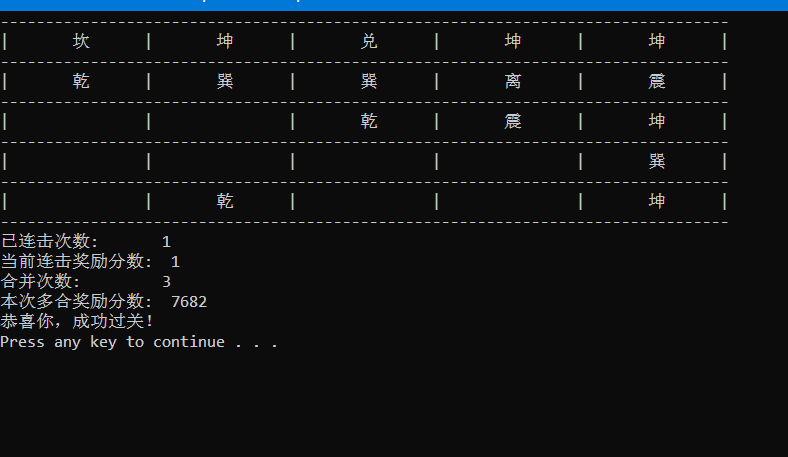


输入完成后进入游戏。

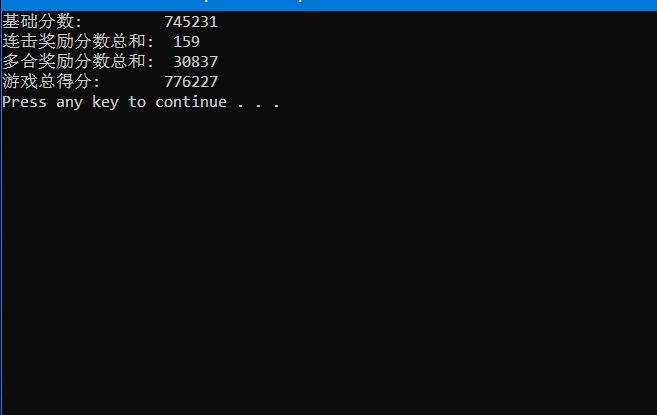


下面为运行一段时间后的成功案例：

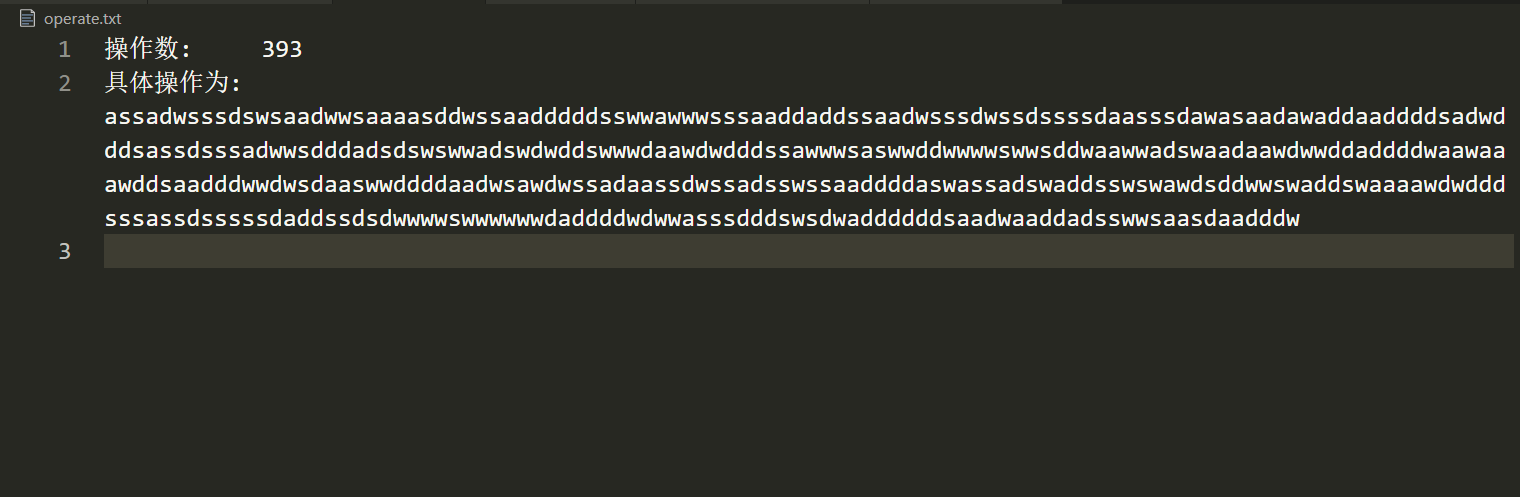
由于采用替换方案的最高数用“兑”代替，故首次出现“兑”即胜利。



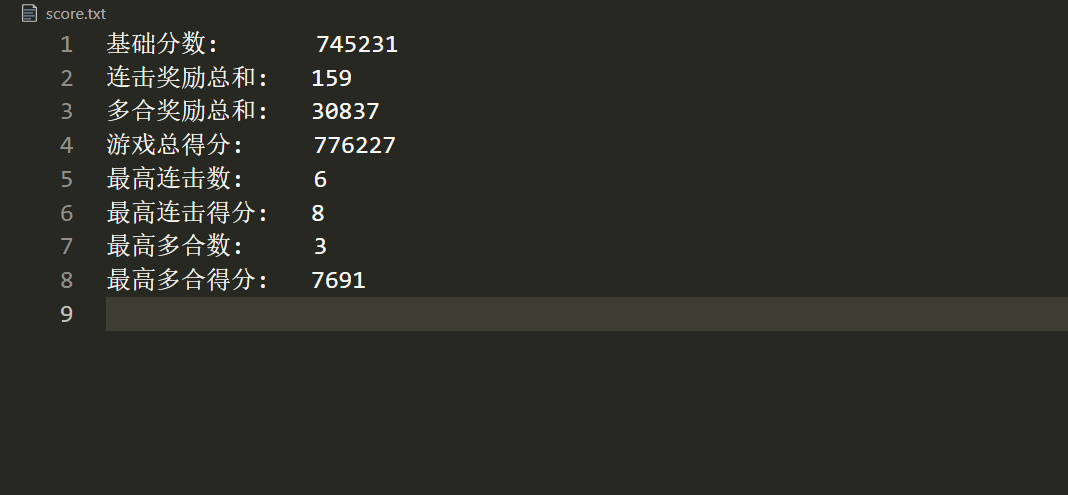
按任意键后显示分数。若再按一次则返回主界面。



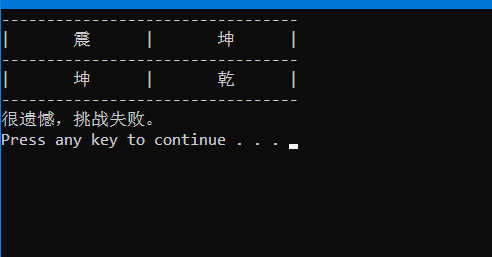
在operate。txt文件中显示操作数和具体操作。



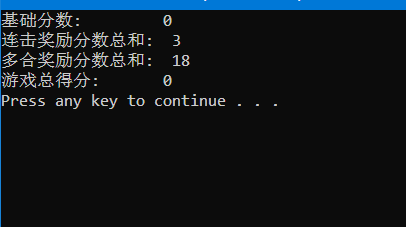
在score.txt文件中显示各种分数。



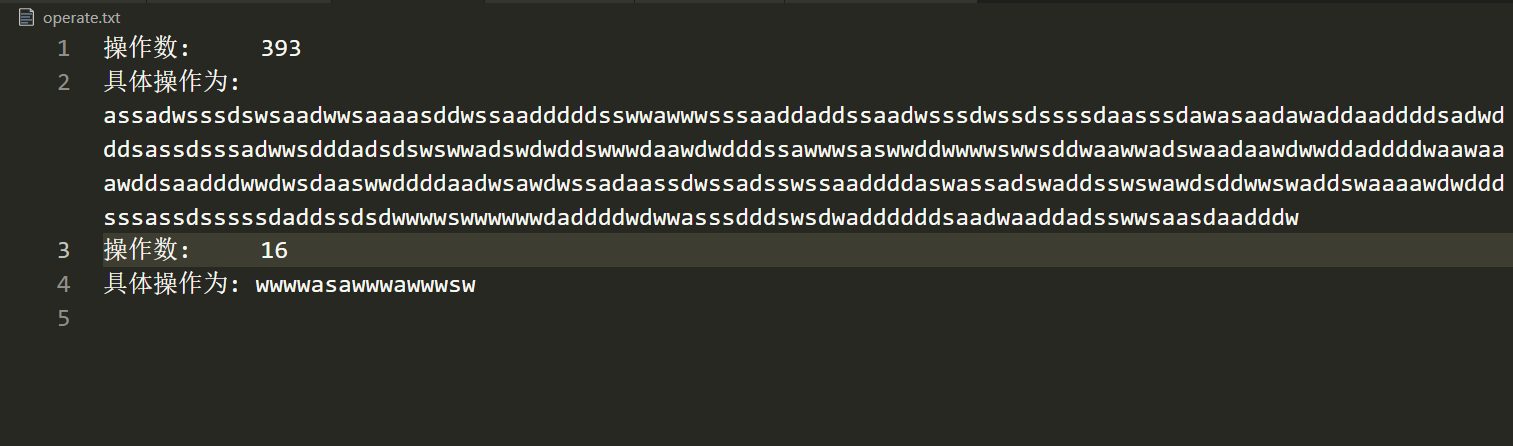
下面为运行一段时间的失败案例：



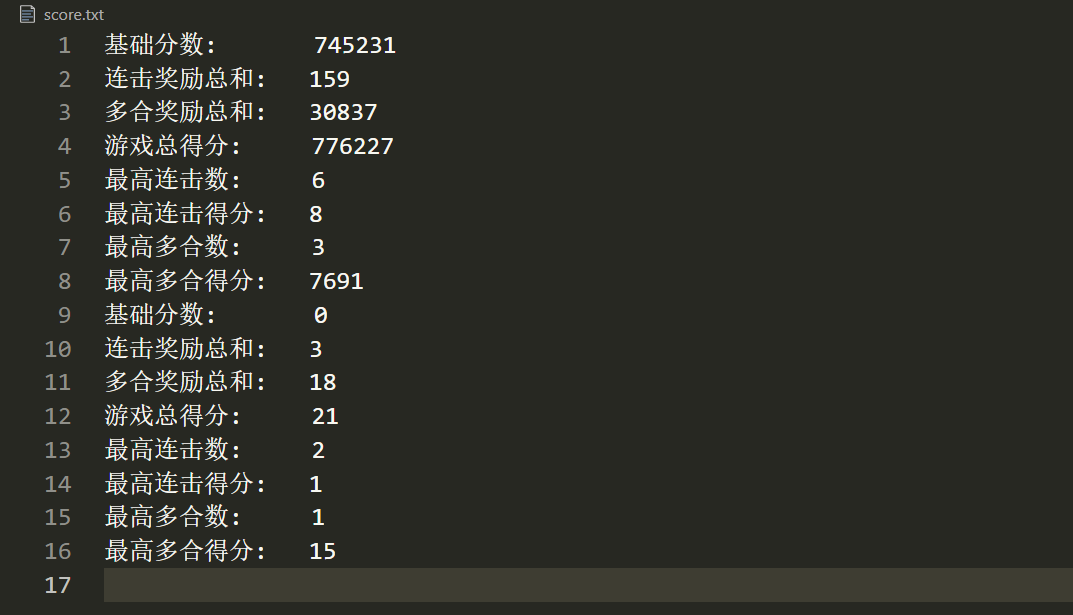
按任意键后显示分数。（再按任意键后返回）



operate.txt文件中继续显示第二次的操作数和具体操作。

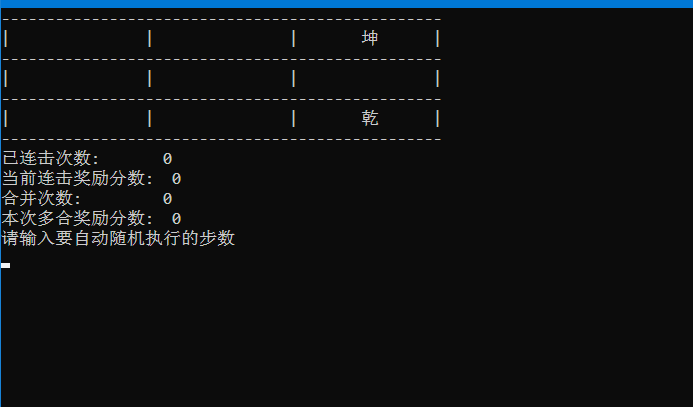


score.txt文件中继续显示各种分数。

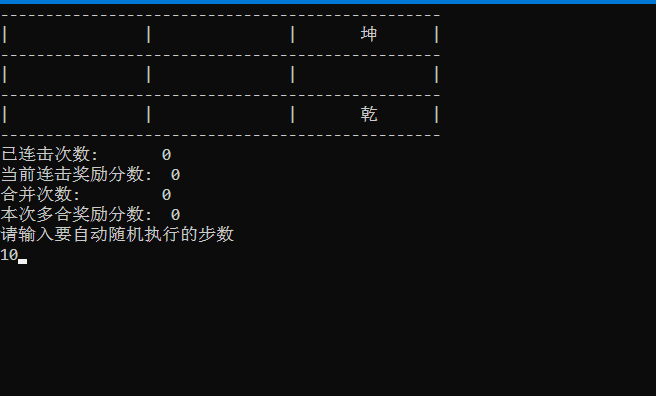


下面为自动运行示例：

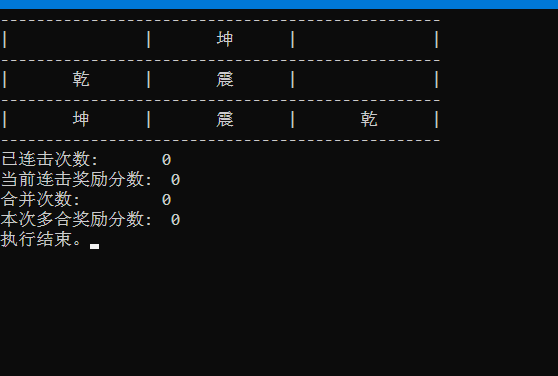
按g后进入自动运行操作。



输入运行步骤数。



运行结束。



3：总结

本次project中涉及到了许多课上没有讲过的内容，例如如何表示汉字字符串，文件之间的通讯等。这些内容需要自行查询，学习。在这个过程中锻炼了我们的自主学习能力和搜索信息的能力。此外，程序中出现的bug也是数不胜数。例如double类型数据的运算，数据的溢出，程序进入死循环等。找到并解决这些bug也是程序设计中的重要一环。再者，函数的封装与调用也是尤为重要的，在面对较大的程序时，在一个main函数中写所有的代码已经不再合适了，这样既不利于分而治之实现问题的简化，也不利于出现错误时快速简洁地找到问题的根源。唯有设计并使用函数才可让主程序既简洁又易于理解，并且大大提高了程序的可移植性。