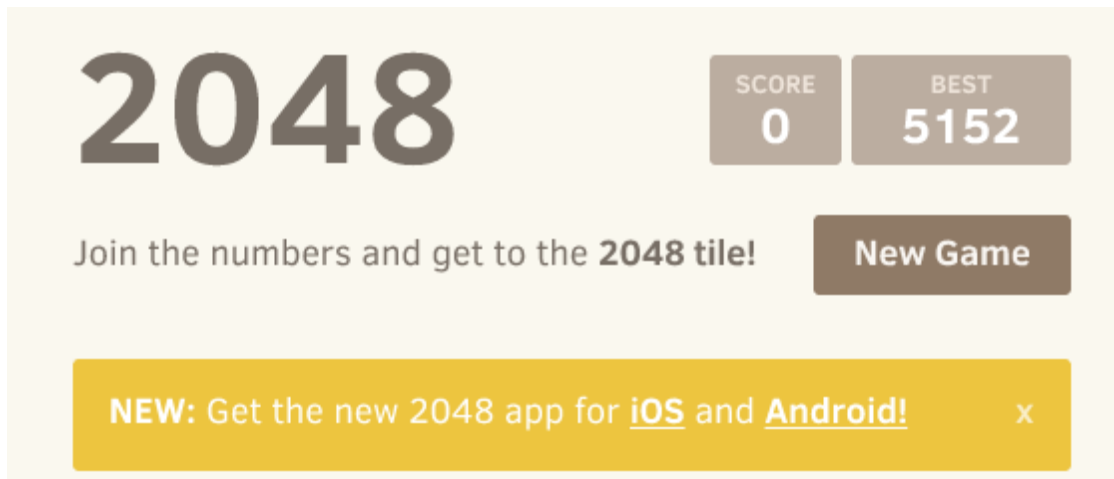


2048 Game

一， 主要函数实现：



根据题目要求需要实现以下几个函数：

moveLeft, moveRight, moveDown, moveUp, gameOver, boardContains2048, printBoard, readBoard ;

题目中表明有些函数体是不能进行修改的，只能添加自己的函数在固定的位置，以及添加所需要的#include，可以改变某些函数体的 return 语句。

其中 **boardContains2048, printBoard, readBoard** 三个函数比较简单，只是二维数组的读入，输出，以及判断二维数组中是否包含 2048；

需要注意的是：为了保证 **printBoard** 函数打印出来的效果，需要提前预估各个符号所占的位数和所在的位置。

printBoard();详见代码，具体需要注意的就是外面那个框是如何保证输出的。

```

void printBoard(int board[SIZE][SIZE]) {
    int i,j;

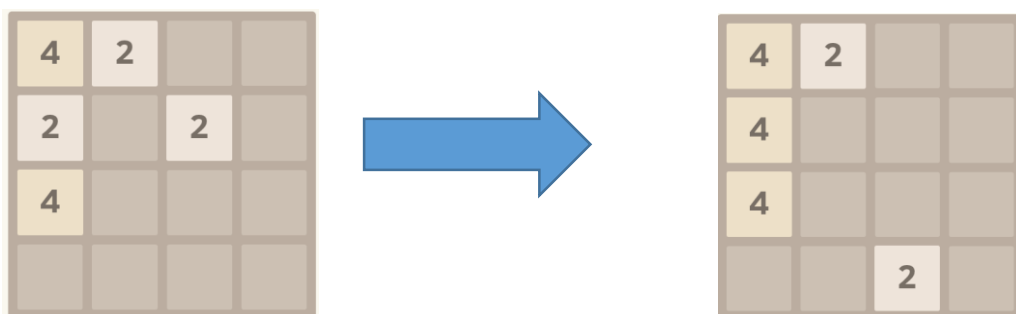
    printf("+");
    for(j=0;j<SIZE;j++)
    {
        printf("----");
    }
    printf("+\n");

    for(i=0;i<SIZE;i++)
    {
        printf("|");
        for(j=0;j<SIZE;j++)
        {
            if(board[i][j]==0)
            {
                char a='.';
                printf("%5c",a);
            }
        }
    }
}

```

二，moveLeft()函数

接下来是比较重要的四个移动函数：上下左右，由于其实现逻辑想通，现在只列出向左移动函数的代码进行解释：



可以发现，向左滑动时，同行相同数字合并，并在边界为没有数字的地方随机产生数字（这个函数即 `InsertNewNumber(board)` 原代码中已经给出了，不需要

我们实现)。

现在我们需要考虑的就是以下问题：

- 1， 向左滑动时，从左侧起，如果有相同元素则合并。
- 2， 需要注意数字 0，即图中的空背景，在滑动的时候是无视的，也就是数字不会被 0 间隔。

上面阐述的或许不是很清楚，需要你多玩玩 2048 本身，就明白我所说的意思了。

所以在代码中我们需要考虑同一行从左到右的列中，两个相邻元素（被 0 间隔也相当于相邻元素）是否相同，相同就叠加并计入分数。

具体分为：

如果当前元素为 0，处理不为 0 的相邻元素，需要把它们移到左边。

如果当前元素不为 0，处理不为 0 的相邻元素，是否需要叠加

代码如下：

```

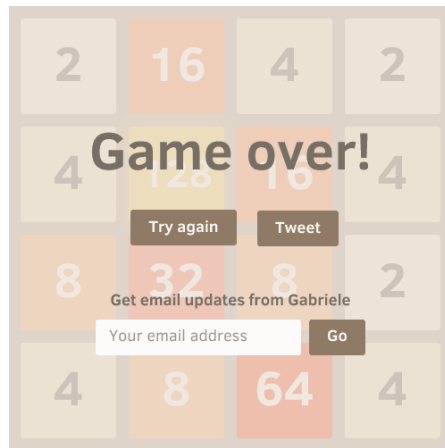
int moveLeft(int board[SIZE][SIZE]) {
    int i,j,score=0,flag=-1;
    for(i=0;i<SIZE;i++)
    {
        for(j=0;j<SIZE;j++)
        {
            int cell=board[i][j];
            if(cell!=0)
            {
                int k=j+1;
                while(k<SIZE)
                {
                    int nextcell=board[i][k];
                    if(nextcell!=0)
                    {
                        if(cell==nextcell)
                        {
                            flag=0;
                            board[i][j]+=board[i][k];
                            score+=board[i][j];
                            board[i][k]=0;
                        }
                    }
                }
            }
        }
    }
}

```

向上、右、下移动情况类似，只需要注意出发点所在的位置即可，不具体描述。

三，gameOver()函数

起初想的比较简单，结果这个函数产生了大量的 bug



主要原因是：

咱们只需要判断当前条件下还能否上下左右移动，如果不能即 `gameOver()`；所以我想当然的认为只需要在 `gameOver()` 函数中添加判断语句判断几个移动函数返回值是否都为-1 即可；

但问题在于，当执行 `if` 语句的时候，它执行了判断（）里面的 `move` 函数，也就是说表面上它只是进行了判断，但实际上它判断的同时，执行了四个移动函数，相当于把输入的 2048 整个已经上下左右移动了一遍。所以最后我依据写的 `move` 函数，采用了全局变量用来标示状态的方法。

```
int gameOver(int board[SIZE][SIZE]) {  
    if(STATE==0)  
    {  
        return 0;  
    }  
    else if(L+R+U+D==-4)  
    {  
        return 1;  
    }  
    else  
    {  
        return 0;  
    }  
}
```

四，一些比较重要的说明

由于每个人代码风格不一样，所以你可能看起来吃力一点，所以你最好提前多玩几遍游戏，然后在纸上面画图分析一下每个方向上移动的逻辑。然后尝试写代码，遇到想不通的可以查看代码。这样效果好点。

进行分析的时候一定要分开各个方向进行单独分析，不要揉杂在一起想。

从代码来看，看懂之后你可以改进以下几个地方：

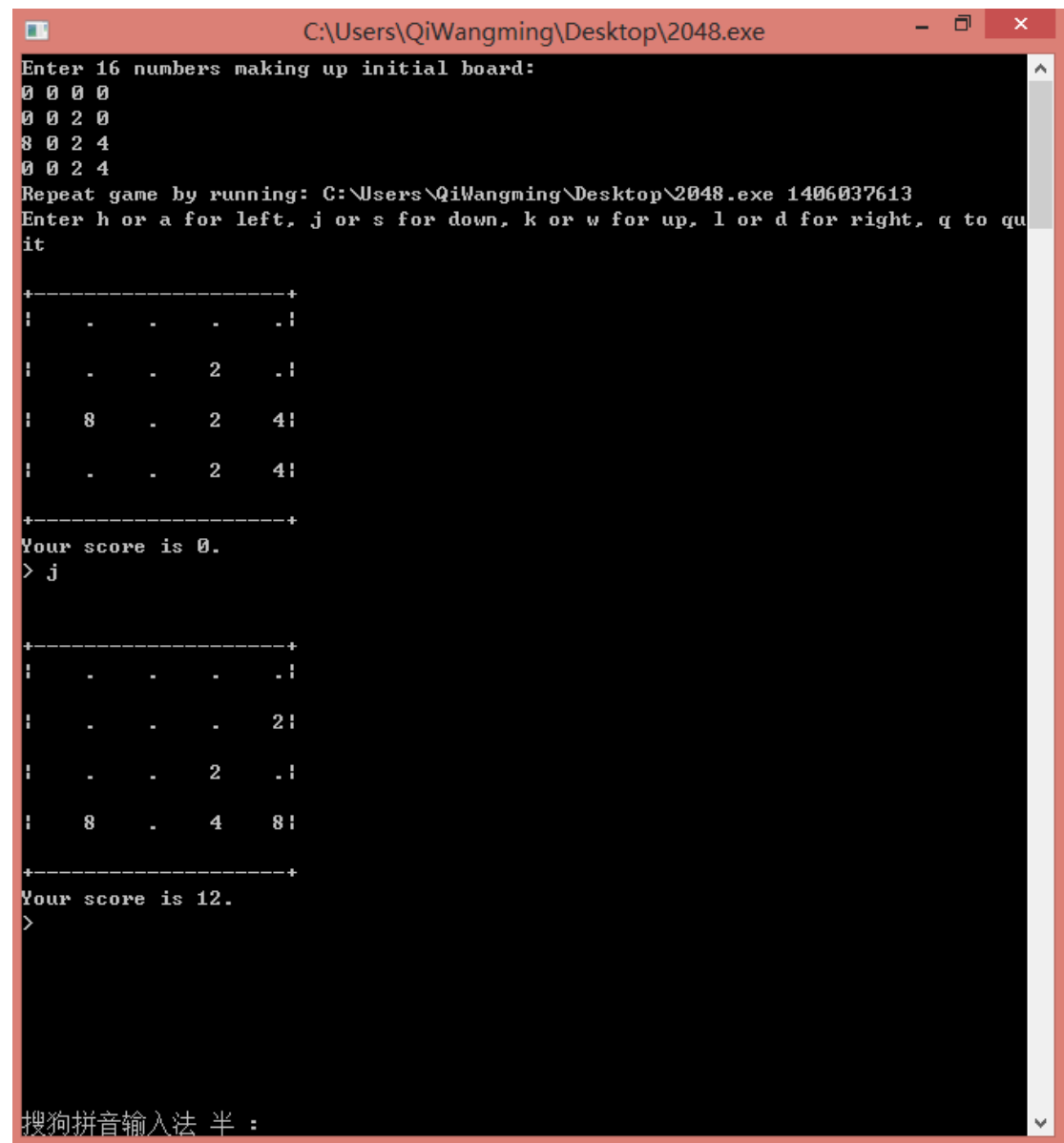
- 1，变量的命名（可以使用有意义的单词）
- 2，关于 `gameOver` 函数的实现，你可以想一个更巧妙的算法。（我写的不太好，但依据我写的 `move` 函数，目前只能采取全局变量的方式）
- 3，游戏逻辑基本都实现了，而且我测试了常见情况和几种极端情况，均没问题，但难免有些数据没经过测试，如果你发现了 `bug` 及时告诉我。

五，关于 diary.txt

你可以按照每天写某一个函数，或者实现某一个移动功能，或者出现哪些 bug，修复 bug 这样的工作量来进行编写。

六，windows 下和 Ubuntu 下运行图

Windows 下的运行截图：



```
C:\Users\QiWangming\Desktop\2048.exe
Enter 16 numbers making up initial board:
0 0 0 0
0 0 2 0
8 0 2 4
0 0 2 4
Repeat game by running: C:\Users\QiWangming\Desktop\2048.exe 1406037613
Enter h or a for left, j or s for down, k or w for up, l or d for right, q to qu
it
+-----+
| . . . . |
| . . 2 . |
| 8 . 2 4 |
| . . 2 4 |
+-----+
Your score is 0.
> j
+-----+
| . . . . |
| . . . 2 |
| . . 2 . |
| 8 . 4 8 |
+-----+
Your score is 12.
>
```

Ubuntu 下的运行截图：

```
qiwangming@ubuntu: ~/桌面
qiwangming@ubuntu:~$ cd 桌面
qiwangming@ubuntu:~/桌面$ g++ -o 2048 2048.c
qiwangming@ubuntu:~/桌面$ ./2048
Enter 16 numbers making up initial board:
0 0 2 0
0 2 2 0
8 2 0 4
0 0 2 4
Repeat game by running: ./2048 1406034873
Enter h or a for left, j or s for down, k or w for up, l or d for right, q to quit
+-----+
| . . 2 . |
| . 2 2 . |
| 8 2 . 4 |
| . . 2 4 |
+-----+
Your score is 0.
> h
+-----+
| 2 . . . |
| 4 . . . |
| 8 2 4 . |
| 2 4 2 . |
+-----+
qiwangming@ubuntu: ~...
Your score is 4.
> j
+-----+
| 2 . . . |
| 2 . . . |
| 8 2 . 4 |
| 2 4 2 . |
+-----+
```

```
qiwangming@ubuntu: ~/桌面
qiwangming@ubuntu:~$ cd 桌面
qiwangming@ubuntu:~/桌面$ ./2048
Enter 16 numbers making up initial board:
0 0 0 0
0 0 2 0
8 2 0 4
2 2 0 4
Repeat game by running: ./2048 1406035035
Enter h or a for left, j or s for down, k or w for up, l or d for right, q to quit
+-----+
| . . . . |
| . . 2 . |
| 8 2 . 4 |
| 2 2 . 4 |
+-----+
Your score is 0.
> j
+-----+
| . . . . |
| . . 2 . |
| 8 . . . |
| 2 4 2 8 |
+-----+
Your score is 12.
> j
+-----+
| . . . . |
| . . 2 . |
| 8 . . . |
| 2 4 2 8 |
+-----+
```