**养正模拟赛**

**普及组**

**竞赛时间：2019 年 5 月 31 日 13:50-16:30**

**OI赛制测试**

**freopen("game.in","r",stdin);**

**freopen("game.out","w",stdout);**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 多边形 | 跳格子 | 2048 | 游戏 |
| 输入文件名 | square.in | jump.in | game.in | store.in |
| 输出文件名 | square.out | jump.out | game.out | store.out |
| 每个测试点时限 | 1s | 1s | 1s | 1s |
| 内存限制 | 64MB | 64MB | 64MB | 64MB |
| 测试点数目 | 10 | 10 | 10 | 10 |
| 测试点分值 | 10 | 10 | 10 | 10 |
| 是否有部分分 | 否 | 否 | 否 | 否 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |

提交源程序必须加后缀

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于 C 语言 | square.c | jump.c | game.c | store.c |
| 对于 C++ 语言 | square.cpp | jump.cpp | game.cpp | store.cpp |
| 对于Pascal语言 | square.pas | jump.pas | game.pas | store.pas |

**最终评测在Linux环境下进行。**

**最终评测忽略行末空格回车。**

**正方形**

**【问题描述】**

又到暑假啦，Conan, Ayumi, Mitsuhiko, Genta一起到Hakase家愉快玩耍（然而你却不行……）。

Ayumi, Mitsuhiko, Genta**依次**开始画正方形（显然Ai是不会参加这个活动的），一人一笔，画完一个，以上一个正方形四边中点为顶点接着画啊画啊画啊……最后一共画了k个正方形。最终他们想知道自己各画了多长。

**【输入格式】**

从文件 ***square.in*** 中读入数据。

输入一行包含两个整数*N,K*，分别表示第一个正方形的边长，正方形个数。

**【输出格式】**

输出到文件 ***square.out*** 中。

输出一行，三个实数，分别表示Ayumi, Mitsuhiko, Genta画的长度，答案**保留3位小数**。

**【样例输入】**

100 2

**【样例输出】**

270.711 241.421 170.711

**【样例解释】**



**【数据规模与约定】**

对于100%的数据满足，。

**跳格子**

**【问题描述】**

大家都说要劳逸结合，Ayumi, Mitsuhiko, Genta画完方格就出去运动啦！

他们来到了一片空地，画了N个连续的方格，每个方格上随机填上了一个数字，大家从第一个格子开始，每次可以向后跳不超过当前格子上的数的步数，大家开始就此比赛，看谁跳到最后一个格子的步数最少。

作为队长的Genta显然是想获得胜利的，所以他打电话给Conan求助，可是Conan在玩游戏，所以就向你求助了。

**【输入格式】**

从文件 ***jump.in*** 中读入数据。

输入第一行包含一个整数*N*，表示画的格子的个数。

第二行包含*N*整数，表示每个格子上的数。

**【输出格式】**

输出到文件 ***jump.out*** 中。

输出一行，表示跳的最少步数。

**【样例输入】**

5

2 3 1 1 1

**【样例输出】**

2

**【数据规模与约定】**

对于40%的数据满足，。

对于100%的数据满足，。

**2048**

**【问题描述】**

此时，Conan却在一旁玩着2048。

这是一个4\*4的矩阵，初始全为0。每次一个没有数字的格子中会随机出现一个2或4，每次可以选择上下左右其中一个方向去滑动，每滑动一次，所有的数字方块都会往滑动的方向靠拢外，相同数字的方块在靠拢、相撞时会相加。

Conan想看看今天自己运气咋样，于是就闭着眼睛，在屏幕上随便滑来滑去。所以这个模拟的任务就交给你了。过了一会，他然后睁开眼睛，如果游戏没有结束（**滑动后如果没有空格子，则游戏结束**），请输出矩阵（**格式参见样例**），**否则输出“Game over!”（不包含引号）**。

**【输入格式】**

从文件 ***game.in*** 中读入数据。

输入第一行包含一个整数*N*，表示Conan滑了几下。

接下来*N* 行，*x, y, v, f*表示第*x*行与第*y*列出现数字为*v*后，Conan滑的方向为*f*（*f*为字符，*U, D, L, R*分别表示向上下左右滑）。

**行从上往下1-4标号，列从左往右1-4标号**。

数据保证在游戏未结束时，只会在空白区域出现数字。

**【输出格式】**

输出到文件 ***game.out*** 中。

输出按题目描述。

**【样例输入】**

8

1 3 4 L

2 3 2 U

2 4 2 R

4 1 2 L

3 4 2 L

3 2 2 D

1 3 4 R

1 3 2 U

**【样例输出】**

0 0 2 8

0 0 0 2

0 0 0 8

0 0 0 0

**【数据规模与约定】**

对于100%的数据满足。

样例补充

输入

4

1 1 2 U

1 2 2 U

1 3 2 U

4 4 2 L

输出

4 2 0 0

0 0 0 0

0 0 0 0

2 0 0 0

输入

4

1 1 2 U

1 2 2 U

1 3 2 U

1 4 2 L

输出

4 4 0 0

0 0 0 0

0 0 0 0

0 0 0 0

**游戏**

**【问题描述】**

“Ran，今天我要在Hakase家打游戏，不回来了。”

“Ran，Hakase新发明了游戏，我今天住博士家。”

“Ran，Conan今天要在我家通宵打游戏。”

终于有一天，电脑被打坏了……2333

所以Conan要前往专卖店买新的，正好专卖店正在促销，一共有三种礼包：

豪华礼包：一个 U盘、一个鼠标和一个机械键盘。

幸运礼包：一个 U盘、两个鼠标。

普通礼包：两个 U盘、一个鼠标。

卖店内准备了 a 个 U盘、b 个鼠标和 c 个机械键盘。为了给顾客带来足够多的惊喜，店长希望相邻两位领礼包的顾客拿到的礼包类型都是不同的。店长想知道这些奖品最多可以发出多少份礼包。可是店长毕竟没有Conan聪明，所以请教Conan，可是Conan要急着回去打游戏，所以就交给你啦。

**【输入格式】**

从文件 ***store.in*** 中读入数据。

输入第一行包含一个正整数 *T*。

接下来*T*行每行包含3个正整数a, b, c，依次表示U盘、鼠标和机械键盘各有多少个。

**【输出格式】**

输出到文件 ***store.out*** 中。

输出*T*行，每行一个整数，表示最多能发出多少份礼包。

**【样例输入】**

2

4 4 0

1 1 1

**【样例输出】**

2

1

**【数据规模与约定】**

对于30%的数据满足，。

对于60%的数据满足，。

对于100%的数据满足，。