

曹文夏令营 2016

普及组 第貳轮

竞赛时间：2016 年 8 月 11 日 8:30-11:30

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, Mitsuhiro, Genta 一起到 Hakase 家愉快玩耍（然而你却不行……）。

Ayumi, Mitsuhiro, 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, Mitsuhiro, Genta 画的长度，答案保留 3 位小数。

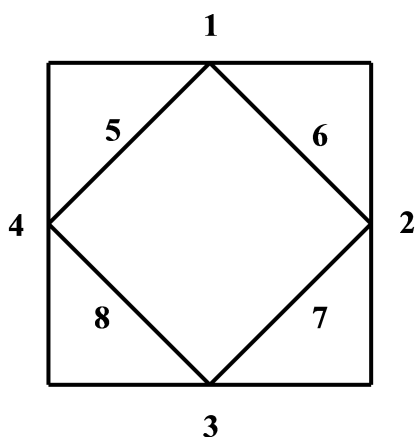
【样例输入】

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 $N \leq 1000$ ， $K \leq 1000$ 。

跳格子

【问题描述】

大家都说要劳逸结合，Ayumi, Mitsuhiro, 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 a_i 。

【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 $N \leq 10$ ， $a_i \leq 10$ 。

对于 100% 的数据满足 $N \leq 5000$ ， $a_i \leq 1000$ 。

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 $N \leq 1000$ 。

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 $T \leq 10$, $a, b, c \leq 30$ 。

对于 60%的数据满足 $T \leq 100$, $a, b, c \leq 300,000$ 。

对于 100%的数据满足 $T \leq 100000$, $0 \leq a, b, c \leq 1,000,000$ 。