

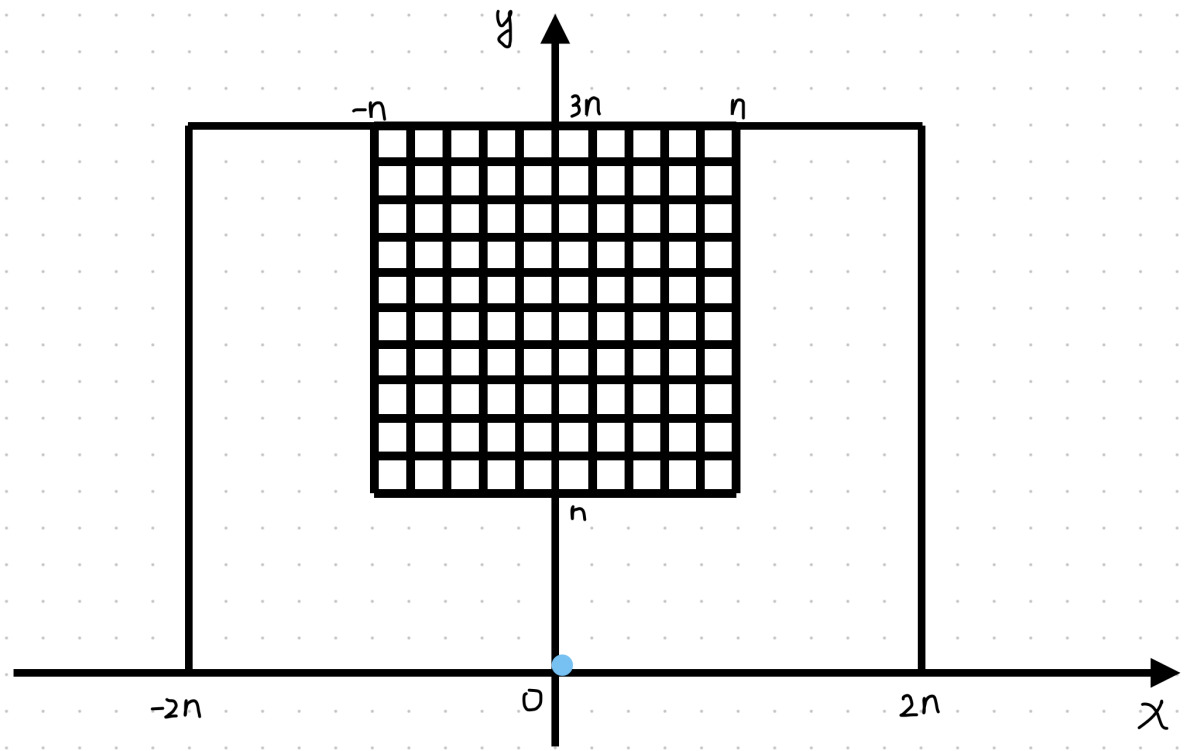
# 机考B卷

## Snooker Brick Out!

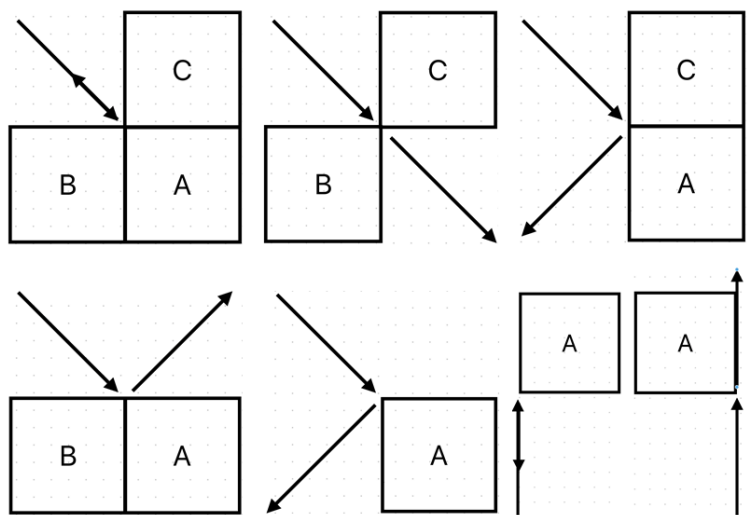
下发文件位于 (<https://github.com/EternalFir/Snooker-Brick-Out-publish>)

### 题目概述

助教闲来无事，发现了一款叫做 *Brick Out* 的小游戏。大致是用一块板控制小球去撞砖块。晚上睡觉的时候，助教梦到了一个 *Brick Out* 的变种。在梦里的游戏中，地图大小为  $4n \times 3n$ ，共有  $2n \times 2n$  个砖块，每个砖块为  $1 \times 1$  的正方形，处于地图上方  $2n$  高度的居中位置。有一个小球，尺寸忽略不计，一开始位于地图下地面中间，有向下为 1，水平方向为 0 的初始速度。



小球每次接触地图底边时，你可以决定让小球的横向速度向左 +2（记为操作 A）或向左 +1（记为操作 B）或不变（记为操作 C）或向右 +1（记为操作 D）或向右 +2（记为操作 E），小球每接触到砖块或者地图边缘，都会在表面上发生一次完全弹性碰撞。当垂直上下运动的时候，认为小球处在网格的节点处且偏右侧。故小球垂直接触到方块的左侧角时认为方块被打掉，发生反弹，右侧角则认为未接触。其他时候，当小球的运动方向发生改变时，认为对应的 A 被打掉。具体碰撞情况详见示例图片，每种情况的旋转都认为情况不变。



若是打掉砖块，则此块砖块消失，得分 +1。大部分砖块为白色，颜色以 0 表示。其中随机分布有一些彩色方块，颜色为 1 – 5。若是连续以 1 – 5 的顺序打掉 5 个方块（中途允许和地图边缘发生碰撞）会获得 50 分的额外技巧加分。请你写一个程序，帮助助教又快又好地打通这个游戏吧！

## 输入

第一行为一个整数  $k$ ，代表测试点编号， $k \in [10]$ 。

第二行为一个整数  $n$ 。

第三行为一个整数  $m$ ，代表允许执行的操作上限。

第四行为一个整数  $s$ ，代表数据中每种颜色的彩色方块的个数。

接下来连续的  $2n$  行，每行有  $2n$  个整数，用空格隔开，代表对应位置方块的颜色。

## 输出

给出连续  $t$  行，每行一个  $A - E$  的大写字母，代表小球每次接触地图下表面后要执行的操作。

**注意：请不要有多余的行**

## 评分

**注意：本题目并非传统题目，你不应该期望得到满分**

每个评测点的评分由两部分组成。分别由游戏得分和控制次数决定。其中游戏得分占 80%，控制次数得分占 20%。游戏得分部分的评分公式为  $\frac{V_{get}}{V_{max}}$ 。其中  $V_{get}$  为运行过程中实际得到的分数，而  $V_{max}$  为理论上的最高得分。控制次数部分的评分公式为  $e^{1-\frac{p}{2n^2}}$ ，其中  $p = \min(t, m)$ ，为模拟器实际上模拟的控制次数。

**注意：若你没能够打掉所有的砖块，则不会得到任何控制分数**

## 备注

**警告：任何试图攻击评测机或者卡bug的行为都会导致零分**

本体一共 10 个评测点，其中 5 个下发，同时还有一个 naive 的样例，不计入 oj 评测，位于 `./testcases`。

保证对 100% 的评测点， $1 \leq n \leq 1 \times 10^3$ ， $m \geq 16n^2$ ，所有彩色砖块的个数相等且彩色方块的总数少于  $n^2$ 。

与题面，数据点同时下发的还有一个本游戏的模拟器，位于 `./Brick-Out-Simulator` 中，且 `./Brick-Out-Simulator/game.h` 的任意代码可以运用到你最终的提交中。具体模拟器用法以及 `Game` 类用法说明详见 `./Brick-Out-Simulator/README.md`。