

# 膜你赛

October 17, 2020

题目名称	糖果游戏	英雄联盟	质数	隔膜
题目类型	传统型	传统型	传统型	传统型
目录	candy	lol	prime	lcyrxc
可执行文件名	candy	lol	prime	lcyrxc
输入文件名	candy.in	lol.in	prime.in	lcyrxc.in
输出文件名	candy.out	lol.out	prime.out	lcyrxc.out
每个测试点时限	1 秒	1 秒	2 秒	1 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	5	10	3	5
测试点是否等分	否	是	否	否

提交源程序文件名

对于 C++ 语言	candy.cpp	lol.cpp	prime.cpp	lcyrxc.cpp
对于 C 语言	candy.c	lol.c	prime.c	lcyrxc.c
对于 Pascal 语言	candy.pas	lol.pas	prime.pas	lcyrxc.pas

编译选项

对于 C++ 语言	-O2 -lm
对于 C 语言	-O2
对于 Pascal 语言	-O2

## 注意事项

1. 需要建立子文件夹。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. 结果比较方式为忽略行末空格、文末回车后的全文比较。
4. C/C++ 中函数 main() 的返回值类型必须是 int，值为 0。

## 糖果游戏 (candy)

### 【题目描述】

Alice 和 Bob 又在玩游戏。

有  $n$  个瓶子，标号分别为  $1, 2, \dots, n$ 。初始状态下第  $i$  个瓶子里有  $a_i$  颗糖果，第  $i$  个瓶子的容量为  $b_i$ 。每个瓶子有一个分裂系数，第  $i$  个瓶子的分裂系数为  $k_i$ 。

游戏的规则如下：Alice 先手，轮到某一个玩家操作时，她可以选择一个瓶子  $i$ ，并选择一种操作：

1. 将第  $i$  个瓶子的糖果数量变为原来的  $k_i$  倍，即  $a_i \leftarrow a_i \cdot k_i$ 。
2. 向第  $i$  个瓶子中添加一颗糖果，即  $a_i \leftarrow a_i + 1$ 。

但是所有操作需要遵循如下规则：每次操作后糖果的数量不能超过该瓶子的容量  $b_i$ ，否则该操作是不被允许的。

如果轮到某一个玩家操作时，不存在允许的操作，则该玩家输掉游戏。

现在，给定所有瓶子的糖果数、容量以及分裂系数，你需要帮她们求出 Alice (先手) 必胜还是 Bob (后手) 必胜。

### 【输入格式】

从文件 *candy.in* 中读入数据。

输入包含多组数据。

第一行包含一个正整数  $T$ ，表示数据组数。

对于每组数据，第一行包含一个正整数  $n$ ，表示瓶子的个数。

接下来  $n$  行，每行三个正整数  $a_i, b_i, k_i$ ，分别表示每个瓶子初始的糖果数、容量以及分裂系数。

### 【输出格式】

输出到文件 *candy.out* 中。

对于每组数据，输出一行，包含一个字符串，如果 Alice (先手) 必胜，则为 Alice，如果 Bob (后手) 必胜，则为 Bob。

### 【样例 1 输入】

```
2
1
3 7 2
2
1 2 2
1 3 3
```

**【样例 1 输出】**

Bob  
Alice

**【样例 1 解释】**

对于第一组数据, Alice 可以将第 1 个瓶子糖果数变为 4 和 6, 此时操作 1 无法进行, 因此无论哪种情况都是 Bob 先让糖果数变成 7 (容量)。

对于第二组数据, Alice 可以向第 2 个瓶子中添加一颗糖果, 于是 Bob 无论在哪个瓶子中添加糖果, Alice 只需在另一个瓶子中添加糖果即可获胜。

**【样例 2】**

见选手目录下的 *candy/candy2.in* 与 *candy/candy2.ans*。

**【数据范围】**

对于所有的测试点, 保证  $1 \leq n, T \leq 10^5; \sum n \leq 10^5; 1 \leq a_i \leq b_i \leq 10^{18}; 2 \leq k_i \leq 10^{18}$ 。

**【子任务】**

- 子任务 1 (8 pts): 保证  $b_i < a_i \cdot k_i$ 。
- 子任务 2 (16 pts): 保证  $b_i \leq 10^5, \sum n \leq 200$ 。
- 子任务 3 (24 pts): 保证  $b_i \leq 10^5$ 。
- 子任务 4 (20 pts): 保证  $k_i = 2$ 。
- 子任务 5 (32 pts): 没有特殊的约束。

## 英雄联盟 (lol)

### 【题目描述】

小  $l$  和小  $c$  在玩 Wild Rift。

众所周知，Wild Rift 中的暴击是非常不靠谱的。

他会一开始设置一个参数  $x\%$ 。如果第一刀不暴，则第二刀的暴率增加到初始值的 2 倍；如果还是不暴，就继续增加到初始值的 3 倍，以此类推，当叠加到 100% 以上的时候，默认下一发一定暴击。

当一次触发了暴击，暴击概率又会重置为  $x$ 。

现在，给你一个  $x\%$ ，要你求出当你砍了  $10^{1000000}$  次刀后，期望暴击了多少刀。

由于答案可能很大，请输出答案除以  $10^{1000000}$  的结果，你的答案和标准答案误差不超过 0.01 就视为正确。

### 【输入格式】

从文件 *lol.in* 中读入数据。

共一行，包含一个正整数  $x$ 。含义详见题面。

### 【输出格式】

输出到文件 *lol.out* 中。

输出一行一个实数，表示答案。

### 【样例 1 输入】

99

### 【样例 1 输出】

1

### 【样例 1 解释】

真实的答案大约是  $\frac{100}{101}$ ，误差不超过 0.01，我们认为是正确的。

### 【数据范围】

本题一共 10 个测试点。

对于第 1 个测试点， $x = 99$ ；

对于第 2 个测试点,  $x = 100$ ;  
对于第 3 个测试点,  $x = 12$ ;  
对于第 4 个测试点,  $x = 25$ ;  
对于第 5 个测试点,  $x = 3$ ;  
对于第 6 个测试点,  $x = 9$ ;  
对于第 7 个测试点,  $x = 3$ ;  
对于第 8 个测试点,  $x = 18$ ;  
对于第 9 个测试点,  $x = 24$ ;  
对于 100% 的数据,  $1 \leq x \leq 100$ 。

### 【后记】

主人公是随（钦）机（定）的。

## 质数 (prime)

### 【题目描述】

小  $\omega$  是个可爱的女孩子, 她特别喜欢质数。

现在她有  $n$  个复数, 分别是  $a_i$  ( $1 \leq i \leq n$ )。

一个数是质数当且仅当它是大于 1 的正整数而且它没有 1 和自己外的正因子。

现在她想让你支持几个操作:

1. 给定  $l, r, x, y$ , 表示将  $a_l$  至  $a_r$  改为  $x + yi$ 。其中  $i$  是虚数单位,  $i^2 = -1$ 。

2. 给定  $l, r, x, y$ , 表示将  $a_l$  至  $a_r$  乘上  $x + yi$ 。其中  $i$  是虚数单位,  $i^2 = -1$ 。

3. 给定  $l, r$ , 询问  $a_l$  至  $a_r$  有几个质数。

可以发现的是:  $(a_0 + b_0i) \times (a_1 + b_1i) = a_0 \times a_1 - b_0 \times b_1 + (a_0 \times b_1 + a_1 \times b_0)i$ 。

### 【输入格式】

从文件 `prime.in` 中读入数据。

第一行两个正整数  $n, q$ , 表示数的个数和操作数。

下面  $n$  行, 每行有两个整数  $u_i, v_i$ , 表示  $a_i = u_i + v_i i$ 。

下面  $q$  行, 每行第一个正整数  $opt_i$  表示操作编号。

若  $opt_i = 1, 2$ , 那么下面紧跟四个整数  $l_i, r_i, x_i, y_i$ , 表示进行一次对应操作。

若  $opt_i = 3$ , 那么下面紧跟两个整数  $l_i, r_i$ , 表示进行一次询问。

### 【输出格式】

输出到文件 `prime.out` 中。

对于每个 3 操作, 输出一行一个非负整数, 表示询问的答案。

### 【样例 1 输入】

```
7 7
0 -2
1 1
1 2
2 0
-2 0
-1 2
0 2
3 2 4
3 3 6
1 5 7 2 -1
```

```
1 4 5 2 -1
3 1 4
1 5 5 -2 2
3 3 3
```

**【样例 1 输出】**

```
1
1
0
0
```

**【样例 2】**

见选手目录下的 *prime/prime2.in* 与 *prime/prime2.ans*。

**【数据范围】**

对于所有的测试点，保证  $1 \leq n, q \leq 2 \times 10^5$ ;  $-5000 \leq x_i, y_i, u_i, v_i \leq 5000$ 。

**【子任务】**

- 子任务 1 (30 pts): 保证  $1 \leq n, m \leq 5000$ 。
- 子任务 2 (30 pts): 保证  $v_i = y_i = 0$ 。
- 子任务 3 (40 pts): 没有特殊的约束。

## 隔膜 (lcyrxc)

### 【题目描述】

小  $l$  和小  $c$  在玩游戏。

她们发明了一个新的游戏。

有一个  $n \times n$  的棋盘，初始有些地方有棋子。

每次行动时，需要在棋盘上找到一个  $k \times k$  的正方形，使得上面没有任何一个棋子。如果无法找到，则她就输掉了这个游戏。

然后，她需要在棋盘任意一个没有棋子的地方放上一个棋子，不需要保证这个棋子在  $k \times k$  的正方形中。

小  $c$  先手，问谁有必胜策略。

### 【输入格式】

从文件 *lcyrxc.in* 中读入数据。

第一行包含两个正整数  $n, k$ 。

接下来  $n$  行，每行输入一个长度为  $n$  的 0/1 字符串，1 表示上面初始有棋子，0 表示上面初始没有棋子。

### 【输出格式】

输出到文件 *lcyrxc.out* 中。

输出一行一个字符串，表示谁赢了。如果小  $l$  赢了，输出 *yc*，否则输出 *rx*。

### 【样例 1 输入】

```
1 1
0
```

### 【样例 1 输出】

```
rx
```

### 【样例 2 输入】

```
1 1
1
```



**【样例 2 输出】**

yc

**【样例 3】**

见选手目录下的 *lcyrxc/lcyrxc3.in* 与 *lcyrxc/lcyrxc3.ans*。

**【数据范围】**

对于 100% 的数据，保证  $1 \leq k \leq n \leq 1000$ 。

**【子任务】**

- 子任务 1 (1 pts): 保证  $k = n$ 。
- 子任务 2 (10 pts): 保证  $2 \times k > n$ 。
- 子任务 3 (19 pts): 保证初始棋盘上没有棋子。
- 子任务 4 (30 pts): 保证  $n \leq 100$ 。
- 子任务 5 (40 pts): 没有特殊的约束。

**【后记】**

文件名、主人公、输出的字符和大样例都是随（钦）机（定）的。  
样例记得缩小看一看哦！