

# T1 Flappy Bird(1s/125M/Flappy.cpp)

## 题目背景

《飞扬的小鸟》是一款风靡的小游戏。

## 题目描述

在游戏中，小鸟一开始位于  $(0, 0)$  处，它的目标是飞到横坐标为  $X$  的某个位置上。

每一秒，你可以选择点击屏幕，那么小鸟会从  $(x, y)$  飞到  $(x + 1, y + 1)$ ，或者不点击，那么小鸟会飞到  $(x + 1, y - 1)$ 。

在游戏中还有  $n$  个障碍物，用三元组  $(x_i, a_i, b_i)$  描述，表示在直线  $x = x_i$  上， $y \leq a_i$  或者  $y \geq b_i$  的部分都是障碍物，碰到或者擦边都算游戏失败。

现在，请你求出小鸟从  $(0, 0)$  飞到目的地最少需要点击多少次屏幕。

## 输入格式

第一行包含两个整数  $n, X$ 。

接下来  $n$  行，每行三个整数  $x_i, a_i, b_i$ 。数据保证  $x_i < x_{i+1}$ 。

## 输出格式

如果无论如何都飞不到目的地，输出 `NIE`，否则输出点击屏幕的最少次数。

## 样例 #1

### 样例输入 #1

```
4 11
4 1 4
7 -1 2
8 -1 3
9 0 2
```

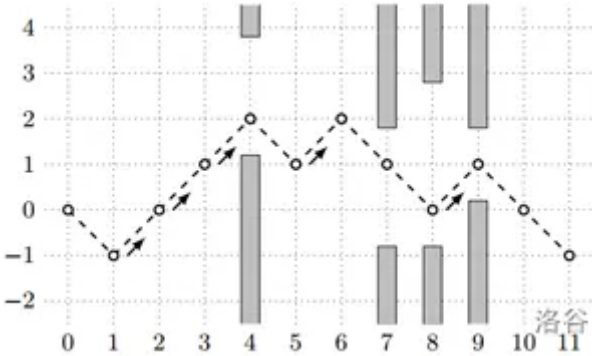
### 样例输出 #1

```
5
```

## 提示

对于 100% 的数据， $0 \leq n \leq 500000$ ， $1 \leq X \leq 10^9$ ， $0 < x_i < X$ ， $-10^9 \leq a_i < b_i \leq 10^9$ 。

### 样例解释：



# T2 Podzielno(1s/125M/Podzielno.cpp)

## 题目描述

$B$  进制数, 每个数字  $i \in [0, B)$  有  $a_i$  个。你要用这些数字组成一个最大的  $B$  进制数  $X$  (不能有前导零, 不需要用完所有数字), 使得  $X$  是  $B - 1$  的倍数。  $q$  次询问, 每次询问  $X$  在  $B$  进制下的第  $k$  位数字是什么 (最低位是第 0 位)。

## 输入格式

第一行包含两个正整数  $B, q$ 。

第二行包含  $B$  个正整数  $a_0, a_1, a_2, \dots, a_{B-1}$ 。

接下来  $q$  行, 每行一个整数  $k$ , 表示一个询问。

## 输出格式

输出  $q$  行, 每行一个整数, 依次回答每个询问, 如果那一位不存在, 请输出  $-1$ 。

## 样例 #1

### 样例输入 #1

```
3 3
1 1 1
0
1
2
```

### 样例输出 #1

```
0
2
-1
```

## 提示

对于 100% 的数据,  $2 \leq B \leq 10^6$ ,  $1 \leq q \leq 10^5$ ,  $1 \leq a_i \leq 10^6$ ,  $0 \leq k \leq 10^{18}$ 。

# T3 Repr(1s/125M/Repr.cpp)

## 题目描述

给定一个数列  $a$ ：

- 当  $n \leq 2$  时,  $a_n = n$ 。
- 当  $n > 2$ , 且  $n$  是奇数时,  $a_n = 2 \times a_{n-1}$ 。
- 当  $n > 2$ , 且  $n$  是偶数时,  $a_n = a_{n-1} + r_{n-1}$ 。

其中  $r_{n-1} = \text{mex}(|a_i - a_j|) (1 \leq i < j \leq n-1)$ ,  $\text{mex}\{S\}$  表示最小的不在  $S$  集合里面的非负整数。

数列  $a$  的前若干项依次为：

1, 2, 4, 8, 16, 21, 42, 51, 102, 112, 224, 235, 470, 486, 972, 990, 1980。

可以证明, 对于任意正整数  $x$ , 只存在唯一一对整数  $(p, q)$  满足  $x = a_p - a_q$ , 定义为  $\text{repr}(x)$ 。

比如  $\text{repr}(17) = (6, 3)$ ,  $\text{repr}(18) = (16, 15)$ 。

现有  $n$  个询问, 每次给定一个正整数  $x$ , 请求出  $\text{repr}(x)$ 。

## 输入格式

第一行包含一个正整数  $n$ 。

接下来  $n$  行, 每行一个正整数  $x$ , 表示一个询问。

## 输出格式

输出  $n$  行, 每行两个正整数  $p, q$ , 依次回答每个询问。

## 样例 #1

### 样例输入 #1

```
2
17
18
```

### 样例输出 #1

```
6 3
16 15
```

## 提示

对于 100% 的数据,  $1 \leq n \leq 10^5$ ,  $1 \leq x \leq 10^9$ 。

# T4 Sabotaż(1s/125M/Sabota.cpp)

## 题目描述

某个公司有  $n$  个人, 上下级关系构成了一个有根树。其中有个人是叛徒(这个人不知道是谁)。

对于一个人, 如果他下属(直接或者间接, 不包括他自己)中叛徒占的比例超过  $x$ , 那么这个人也会变成叛徒, 并且他的所有下属都会变成叛徒。你要求出一个最小的  $x$ , 使得最坏情况下, 叛徒的个数不会超过  $k$ 。

## 输入格式

第一行包含两个正整数  $n, k$ 。

接下来  $n - 1$  行, 第  $i$  行包含一个正整数  $p_{i+1}$ , 表示  $i + 1$  的父亲是  $p_{i+1}$ 。

## 输出格式

输出一行一个实数  $x$ , 误差在  $10^{-6}$  以内都被认为是正确的。

所以本题使用 SPJ。

## 样例 #1

### 样例输入 #1

```
9 3
1
1
2
2
2
3
7
3
```

### 样例输出 #1

```
0.6666666667
```

## 提示

对于 100% 的数据,  $1 \leq k \leq n \leq 500000, 1 \leq p_{i+1} \leq i$ 。

### 样例说明:

答案中的  $x$  实际上是一个无限趋近于  $\frac{2}{3}$  但是大于  $\frac{2}{3}$  的数。

因为当  $x$  取  $\frac{2}{3}$  时, 最坏情况下 3, 7, 8, 9 都是叛徒, 超过了  $k = 3$ 。

# T5 One Occurrence(3s/750M/occur.cpp)

## 题目描述

给定一个数组  $a$ ，长度为  $n$ ，以及  $q$  个询问。第  $i$  个询问由两个整数  $l_i$  和  $r_i$  表示。对于每个询问，你需要找到在  $a$  的子数组（从下标  $l_i$  到  $r_i$ ）中仅出现一次的数字之一。如果不存在这样的数字，则输出 0。

例如：

- 如果  $a = [1, 1, 2, 3, 2, 4]$ ：
  - 对于询问  $(l_i = 2, r_i = 6)$ ，子数组为  $[1, 2, 3, 2, 4]$ ，可能的答案有 1, 3 和 4。
  - 对于询问  $(l_i = 1, r_i = 2)$ ，子数组为  $[1, 1]$ ，不存在仅出现一次的数字。

你能回答所有的询问吗？

## 输入格式

第一行包含一个整数  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ )。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 5 \cdot 10^5$ )。

第三行包含一个整数  $q$  ( $1 \leq q \leq 5 \cdot 10^5$ )。

接下来  $q$  行，每行包含两个整数  $l, r$  ( $1 \leq l \leq r \leq n$ )，表示一次询问。

## 输出格式

对于每个询问：

- 如果区间内不存在只出现一次的数，输出 0；
- 否则，输出任意一个只出现一次的数。

## 样例 #1

### 样例输入 #1

```
6
1 1 2 3 2 4
2
2 6
1 2
```

### 样例输出 #1

```
4
0
```