

# NOI Online普及组试题

## 题目概览

题目名称	王国比赛	数学游戏	字符串
题目类型	传统题	传统题	传统题
目录	kingdom	math	string
可执行文件名	kingdom	math	string
输入文件名	kingdom.in	math.in	string.in
输出文件名	kingdom.out	math.out	string.out
每个测试点时限	1.0s	1.0s	2.0s
内存限制	256MB	256MB	512MB
子任务数目	10	20	20
测试点是否等分	是	是	是

## 提交源程序文件名

对于 C++ 语言	kingdom.cpp	math.cpp	string.cpp
-----------	-------------	----------	------------

## 编译选项

对于 C++ 语言	-lm -O2 -std=c++14
-----------	--------------------

## 注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 程序可使用的栈空间内存限制与题目的内存限制一致。
5. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。  
上述时限以此配置为准。
6. 只提供 Linux 格式附加样例文件。
7. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

# 王国比赛(kingdom)

## 题目描述

智慧之王 Kri 统治着一座王国。

这天 Kri 决定举行一场比赛，来检验自己大臣的智慧。

比赛由  $n$  道判断题组成，有  $m$  位大臣参加。现在你已经知道了所有大臣的答题情况，但尚未拿到答案，于是你决定先行预测。

具体来说，对于第  $i$  道题，有  $x$  个大臣选对， $y$  个大臣选错（显然有  $x + y = m$ ），如果  $x > y$ ，那么你预测这题答案为对，否则为错。为了方便，我们保证  $m$  是奇数。

在统计完成后，你拿到了答案，你想知道通过你的预测方式你最后有几道题预测正确。

## 输入格式

第一行两个正整数  $n, m$ ，保证  $m$  是奇数。

接下来  $m$  行，每行  $n$  个整数，第  $i$  行第  $j$  个整数  $a_{ij}$  代表第  $i$  位大臣对第  $j$  道题的答案，1 表示他选对，0 表示他选错。

接下来 1 行  $n$  个整数，表示比赛答案，第  $i$  个数  $b_i$  若为 1 表示第  $i$  道题答案是对，若为 0 表示答案是错。

## 输出格式

输出一个整数，表示你最后有几题预测正确。

## 输入输出样例

### 样例 1 输入

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

### 样例 1 输出

```
2
```

### 样例 1 解释

- 第一题  $x = 1, y = 2$  你预测答案为错（即0），实际答案为1，预测错误。
  - 第二题  $x = 2, y = 1$  你预测答案为对（即1），实际答案为1，预测正确。
  - 第三题  $x = 2, y = 1$  你预测答案为对（即1），实际答案为1，预测正确。
- 所以预测正确的题数为2。

## 样例 2 输入

```
5 6
1 0 1 1 1 0
0 1 0 1 1 1
0 0 1 0 1 0
1 0 1 0 1 0
0 1 0 1 0 0
1 0 1 0 1 0
```

## 样例 2 输出

```
4
```

## 附加样例

见样例目录下的 kingdom3.in 和 kingdom3.out 。

## 数据范围

对于 20% 的数据,  $n \leq 5, m = 1$  。

对于 50% 的数据,  $n \leq 10, m \leq 10$  。

对于 100% 的数据,  $n \leq 1000, m \leq 1000$ ,  $m$  为奇数 。

# 数学游戏(math)

## 题目描述

Kri 喜欢玩数字游戏。

一天，他在草稿纸上写下了  $t$  对正整数  $(x, y)$ ，并对于每一对正整数计算出了  $z = x \times y \times \gcd(x, y)$ 。

可是调皮的 Zay 找到了 Kri 的草稿纸，并把每一组的  $y$  都擦除了，还可能改动了一些  $z$ 。

现在 Kri 想请你帮忙还原每一组的  $y$ ，具体地，对于每一组中的  $x$  和  $z$ ，你需要输出最小的正整数  $y$ ，使得  $z = x \times y \times \gcd(x, y)$ 。如果这样的  $y$  不存在，也就是 Zay 一定改动了  $z$ ，那么请输出  $-1$ 。

注： $\gcd(x, y)$  表示  $x$  和  $y$  的最大公约数，也就是最大的正整数  $d$ ，满足  $d$  既是  $x$  的约数，又是  $y$  的约数。

## 输入格式

第一行一个整数  $t$ ，表示有  $t$  对正整数  $x$  和  $z$ 。

接下来  $t$  行，每行两个正整数  $x$  和  $z$ ，含义见题目描述。

## 输出格式

对于每对数字输出一行，如果不存在满足条件的正整数  $y$ ，请输出  $-1$ ，否则输出满足条件的最小正整数  $y$ 。

## 输入输出样例

### 样例1输入

```
1
10 240
```

### 样例1输出

```
12
```

### 样例 1 解释

$$x \times y \times \gcd(x, y) = 10 \times 12 \times \gcd(10, 12) = 240$$

### 样例 2 输入

```
3
5 30
4 8
11 11
```

## 样例 2 输出

```
6
-1
1
```

## 附加样例

见样例目录下的 math3.in 和 math3.out , 以及 math4.in 和 math4.out 。

## 数据范围

对于 20% 的数据,  $t, x, z \leq 10^3$ 。

对于 40% 的数据,  $t \leq 10^3, x \leq 10^6, z \leq 10^9$ 。

对于另 30% 的数据,  $t \leq 10^4$ 。

对于另 20% 的数据,  $x \leq 10^6$ 。

对于 100% 的数据,  $1 \leq t \leq 5 \times 10^5, 1 \leq x \leq 10^9, 1 \leq z < 2^{63}$ 。

# 字符串(string)

## 题目描述

Kri 非常喜欢字符串，所以他准备找  $t$  组字符串研究。

第  $i$  次研究中，Kri 准备了两个字符串  $S$  和  $R$ ，其中  $S$  长度为  $n$ ，且只由  $0$ ， $1$ ， $-$  三种字符构成（注：这里的第三种字符是减号）， $R$  初始时空。

每次研究，Zay 会带着一个美丽的长度为  $m$  的字符串  $T$  来找 Kri 玩，Kri 非常羡慕 Zay 拥有如此美丽的字符串，便也想用字符串  $S$  和  $R$  变出字符串  $T$ 。

具体地，Kri 将会进行  $n$  次操作。每次操作中，Kri 会取出  $S$  的第一个字符（记为  $c$ ），并将其从  $S$  中删去。如果  $c = -$ ，则 Kri 要删去  $R$  的开头字符或结尾字符（数据保证删去后  $R$  不为空）。否则，Kri 会将  $c$  加入到  $R$  的末尾。

当进行完所有操作后，Kri 会检查  $R$  是否和  $T$  相等。如果  $R = T$ ，Kri 就会感到开心；否则，Kri 会感到难受。

请问在每次研究中，Kri 有多少种操作方式使自己最后感到开心？我们定义两种方案不同，当且仅当在某种方案的某次操作中，Kri 删去了  $R$  的开头字符。而在另一种方案的这次操作中，Kri 删去了  $R$  的结尾字符。

由于答案可能很大，你只需要输出答案除以  $1,000,000,007$ （即  $10^9 + 7$ ）的余数。

## 输入格式

第一行一个正整数  $t$ 。

接下来有  $t$  组数据分别表示  $t$  次字符串的研究，对于每组数据：

第一行有两个正整数  $n, m$ ，分别表示字符串  $S, T$  的长度。

第二行是字符串  $S$ 。

第三行是字符串  $T$ 。

## 输出格式

共  $t$  行，第  $i$  行表示第  $i$  组研究的答案。

## 输入输出样例

### 样例1输入





```
3
6 2
10-01-
01
7 3
010-1-1
101
6 4
111-00
1100
```

## 样例1输出

```
2
1
2`-1`
```

## 样例 1 解释

对于第一组数据，有以下两种方案：

- 第一个  删  $R$  的开头，第二个  删  $R$  的结尾。
- 第一个  删  $R$  的结尾，第二个  删  $R$  的开头。

## 附加样例

见样例目录下的 `string2.in` 和 `string2.out` 。

## 数据范围

对于 20% 的数据， $n, m \leq 15$ 。

对于 30% 的数据， $n, m \leq 30$ 。

对于 70% 的数据， $n, m \leq 80$ 。

对于另 10% 的数据，保证答案不超过 1。

对于 100% 的数据， $1 \leq t \leq 5, 1 \leq n, m \leq 400$ 。