

题目名称	graph	gcds	fight	perm
题目类型	传统型	传统型	传统型	传统型
输入文件名	<i>graph.in</i>	<i>gcds.in</i>	<i>fight.in</i>	<i>perm.in</i>
输出文件名	<i>graph.out</i>	<i>gcds.out</i>	<i>fight.out</i>	<i>perm.out</i>
子任务个数	3	3	无	7
测试点个数	50	60	20	47
时间限制	1s	1s	1s	2s
空间限制	128MiB	128MiB	128MiB	64MiB
源程序文件名 (C++)	<i>graph.cpp</i>	<i>gcds.cpp</i>	<i>fight.cpp</i>	<i>perm.cpp</i>

编译选项 (C++) : `-lm -std=c++17 -O2`

若无特殊说明，比对方式为逐行比对（忽略行末空格、制表符和文末回车）。

数据范围内的表格留空表示无特殊限制。

难度顺序与题目顺序无关

A (graph)

题目描述

有一个 n 个点的竞赛图（即任意不同两点间恰有一条边的有向简单图）。

接下来有 m 次操作，每次给定两个不同的点 u, v ，然后将 u, v 之间的边反向（即，若原来这条边是从 u 指向 v 则改为 v 指向 u ，反之亦然）。

如果在整个过程中，这个竞赛图没有成为过强连通图（即，任意时候，存在 u, v 使得从 u 开始没有到 v 的路径），那么我们称这个图是好的。

现在给定了 n, m 和这 m 次操作的 u, v 。请构造出一个好的竞赛图。

若有多个好的竞赛图，输出任意一个均算对。可以证明在本题条件下有至少一个好的竞赛图。

输入格式

从文件 *graph.in* 中读入数据。

第一行，两个正整数 n, m 。

接下来 m 行，第 i 行两个不同的正整数 u, v ，表示第 i 次操作的点对。

输出格式

输出到文件 *graph.out* 中。

输出 $n - 1$ 行。第 i 行包含 $n - i$ 个数，值均为 0 或 1。

对于任意点对 $1 \leq u < v \leq n$ ，若 u, v 间的边为 u 指向 v ，则第 u 行第 $v - u$ 个数是 1，否则是 0。

样例

样例输入

```
6 4
1 3
3 4
2 6
5 6
```

样例输出

```
1 0 1 1 1
0 0 0 1
1 1 1
1 1
0
```

数据范围

本题比对方式为自定义校验器 (special judge)

对于所有数据，有 $1 \leq n \leq 400, 1 \leq m \leq \frac{n(n-1)}{2}$ 。

子任务编号	分值	$m \leq$
1	30	$n - 2$
2	40	$\lceil \frac{3n}{2} \rceil - 3$
3	30	$2n - 14$

B (gcds)

题目描述

给定长为 n 的正整数序列 a ，求可以修改 a 中的任意一个元素，使它变为 $\leq 5 \cdot 10^5$ 的任意一个正整数时，它的 $\gcd > 1$ 的子串数量的最大值。

输入格式

从文件 *gcds.in* 中读入数据。

第一行，一个正整数 n 。

第二行， n 个正整数 a_i 。

输出格式

输出到文件 *gcds.out* 中。

一行，一个正整数，表示答案。

样例 1

样例输入

```
5
4 5 10 3 7
```

样例输出

```
9
```

样例解释

对于样例 1，其中一种方案是：将 a_4 改为 70，此时 $\gcd > 1$ 的子串有 $(4), (5), (10), (70), (7), (5, 10), (10, 70), (70, 7), (5, 10, 70)$ ，共 9 个。可以证明不存在使答案更大的方案。

样例 2

见选手目录下的 *ex_data2.in* 和 *ex_data2.out*，此样例满足子任务 2 的限制。

数据范围

对于所有数据， $n \leq 5 \cdot 10^4$, $a_i \leq 5 \cdot 10^5$ 。

子任务编号	分值	$n \leq$	$a_i \leq$	特殊性质	子任务依赖
1	15	100	100	✓	
2	35		500	✓	1
3	50				2

特殊性质： a_i 在范围内等概率随机生成。

C (fight)

题目描述

Yqh 的梦想是成为一名近战战士，在队友的增幅下冲锋陷阵。

Yqh 的队友 **Dr.W** 是一名牧师，他有两种法术：

1. 增幅法术，使得 **Yqh** 的战斗力的增加 1，但人类是有极限的，所以当 **Yqh** 的战斗力等于 MX 时，该操作无效。
2. 削弱法术，使得所有敌人和队友的战斗力的减少 1，当然，一个人的战斗力不能小于 0。所以当 **Yqh** 的战斗力的等于 0 时，该操作无效。

我们现在已经知道了 **Dr.W** 的施法顺序，假如 **Yqh** 的初始战斗力为 V_0 时，不难得出她最后的战斗力 V_t 。

然而，对于 **Yqh** 而言，如果她过长时间没有接受到 **Dr.W** 的法术，她将获得一层法术抗性并使得 **Dr.W** 发的下一次法术对她无效。

形式化的来说，我们规定一个阈值 W ，当一个法术的使用时间 - 上一个法术的使用时间 $\leq W$ ，则这个法术没有效果（但这个法术仍然会认为被使用了）

现在，你要求出一个最大的 W ，使得存在一个 V_0 ， $V_t = V_S$ 。并求出在 W 最大时，使得 $V_t = V_S$ 的 V_0 的数量，其中 V_S 是我们给定的一个值。

输入格式

从文件 *fight.in* 中读入数据。

第一行三个整数 n, MX, V_S 。

接下来 n 行，每一行为一个字符 op 和一个整数 T_i 。

当 $op = +$ 时，表示在 T_i 时刻时，**Dr.W** 释放了一次增幅法术。

当 $op = -$ 时，表示在 T_i 时刻时，**Dr.W** 释放了一次削弱法术。

输出格式

输出到文件 *fight.out* 中。

一行，两个整数，表示最大的 W 和 V_0 的数量。

样例

见选手目录下的 *ex_fight1.in* 和 *ex_fight1.out*。

数据范围

对于 50% 的数据, $n \leq 4000$ 。

对于 55% 的数据, $MX \leq 4$ 。

对于 100% 的数据, $n \leq 100000, MX \leq 5000, 0 \leq T_i \leq 10^9$ 。

D (perm)

题意描述

下文中「排列」均指 1 到 n 的排列。

两个排列 A, B 保证 $\forall i, A_i \neq B_i$ 。求有多少个排列 C 满足 $\forall i, C_i \neq A_i, C_i \neq B_i$ 。对 1000000007 取模。

输入格式

第一行一个正整数 T 表示数据组数。

对于每组数据，三行，分别为 n, A, B 。

输出格式

对于每组数据，一行一个整数表示答案。

样例 1

样例输入

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

样例输出

```
4
```

样例解释

合法的排列是 $\{3, 1, 4, 2\}, \{3, 2, 4, 1\}, \{4, 1, 3, 2\}, \{4, 2, 3, 1\}$ 。

样例2~8

见选手目录下的 `ex_perm?.in` 和 `ex_perm?.out`。

数据范围

对于所有数据，保证 $1 \leq T \leq 10, 2 \leq n \leq 1000, A_i \neq B_i$ 。

子任务编号	分值	$n \leq$	特殊性质	子任务依赖
1	2	10		
2	5	20		1
3	13	50		2
4	19	300		3
5	7		$A_i = B_{i \bmod n+1}$	
6	7		$\exists k n, B_i = A_{i-1+k[i \bmod k=1]}$	
7	47			4, 5, 6