

## 问题 A.澳大利亚

Pigeland 大学将主办 2224 年动物高校程序设计竞赛（ACPC 2224）。与往年不同的是，主办院校的参赛队都被标记为非官方参赛队，而这次 Pigeland 将派出正式参赛队，每支参赛队将配备三台工作站。然而，Pigeland 大学教练 Pig-head 仍然不确定他的队伍能否获得金牌。因此，他决定以需要向在线裁判上传数据为借口，提前从澳大问题设置小组那里获取竞赛问题。

为防止作弊，AUS 尝试使用特殊密码对问题进行加密。具体来说，问题由小写英文字母组成的字符串表示。AUS 希望设计一个将小写英文字母映射为小写英文字母的密码函数  $f(x)$ 。对于问题  $S = s_1s_2 \dots s_n$ ，问题的加密版本是另一个字符串，其值为  $F(S) = f(s_1)f(s_2) \dots f(s_n)$ 。当  $S = \text{abcabc}$  和  $f(a) = a$ ,  $f(b) = k$ ,  $f(c) = a$  时，加密版本为  $F(S) = \text{akaaka}$ 。

作为 AUS 的成员，你的任务是设计密码函数  $f$ 。AUS 的领导者认为，当且仅当存在至少一个问题可以加密成与另一个问题相同的加密版本，而不是所有问题都产生相同的加密输出时，函数才是强函数。为了验证这一点，他会给你三个问题  $S_1$ 、 $S_2$  和  $S_3$ ，你需要找到一个密码函数  $f$ ，使得  $F(S_1) = F(S_2)$  和  $F(S_1) \neq F(S_3)$ 。由于 AUS 有几位经验丰富的成员，您的任务只是确定是否存在这样的密码函数。

### 输入

有多个测试用例。第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行包含一个仅由小写英文字母组成的字符串  $S_1$  ( $1 \leq |S_1| \leq 10^3$ )。第二行包含一个仅由小写英

文字母组成的字符串  $S_2$  ( $1 \leq |S_2| \leq 10^3$ )。第三行包含一个仅由小写英文字母组成的字符串  $S_3$  ( $1$

$\leq |S_3| \leq 10^3$ )。保证  $|S_1| + |S_2| + |S_3|$  所有测试用例的总和不超过  $3 \times 10^4$ 。

### 输出

对于每个测试用例，如果存在此类密码函数，则在一行中输出 "是"。否则，输出 NO。

### 示例

标准输入	标准输出
4	是
abab	没有
cdcd	是
弊	没有
abab	
cdcd	
abcd	
abab	
cdcd	
abc	

地区比赛 yz 捍卫	
------------------	--

备注

对于第一和第三个样本测试用例，一个有效的密码函数可以是 $f(a)=f(b)=f(c)=f(d)=a$ 和 $f(e)=b$ 。

地区比赛

问题 B. 巴克利 III

小猪王国里有  $n$  只小猪。它们都精通编程竞技，其中第  $i$  只的评分为  $a_i$ 。如果  $k$  只小猪  $p_1, p_2, \dots, p_k$  组成一个团队，则该团队的等级为  $a_{(p)(1)} \& a_{(p)(2)} \& a_{(p)(3)} \& \dots \& a_{(p)(k)}$ ，其中  $\&$  表示位和运算。

将举行一些编程比赛。每场比赛，猪场可以派出一支队伍参赛。在第  $i$  次比赛中，只有编号在  $l_i$  和  $r_i$  之间（包括  $l_i$  和  $r_i$ ）的猪才有时间参赛。不幸的是，由于资金短缺，编号在  $l_i$  和  $r_i$  之间的一头猪必须离开。与此同时，区间内的所有其他猪都将参加比赛。猪地的教练猪头需要正确选择不参加比赛的猪，这样才能最大限度地提高团队的评分。

不过，通过训练和参加比赛，猪的等级可能会发生变化。作为猪头的好朋友，您的任务是在以下三种类型的  $q$  个活动中保持猪的等级。

- 1  $l\ r\ x$ : 猪头通过执行与  $x$  的位和运算，改变编号在  $l$  和  $r$  之间（包括  $l$  和  $r$ ）的每头猪的评级。更正式地说，对于所有  $l \leq i \leq r$ ， $a_i$  变成  $a_i \& x$ 。
- 2  $s\ x$ : 猪头将第  $s$  只猪的等级改为  $x$ 。
- 3  $l\ r$ : 猪头在组队时要求获得最高分，方法是选择编号在  $l$  和  $r$  之间（包括  $l$  和  $r$ ）的猪，并恰好去掉其中一只。

输入

每个测试文件中只有一个测试用例。

第一行包含两个整数  $n$  和  $q$  ( $2 \leq n \leq 10^6, 1 \leq q \leq 10^6$ )，表示猪的数量和事件的数量。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 2^{63}$ )，其中  $a_i$  表示第  $i$  头猪的等级。

在以下  $q$  行中，第  $i$  行首先包含一个整数  $op_i$  ( $op_i \in \{1, 2, 3\}$ )，表示第  $i$  个事件的类型。如果  $op_i = 1$ ，则后面有三个整数  $l, r$  和  $x$  ( $1 \leq l \leq r \leq n, 0 \leq x < 2^{63}$ )；如果  $op_i = 2$ ，则后面有两个整数  $s$  和  $x$  ( $1 \leq s \leq n, 0 \leq x < 2^{63}$ )；如果  $op_i = 3$ ，则有两个整数  $l$  和  $r$  ( $1 \leq l < r \leq n$ )。

输出

对于第三种类型的每个事件，输出一行，其中包含一个整数，表示团队的最高评级。

示例

标准输入	标准输出
5 9	7
7 7 7 6 7	6
3 1 5	7
2 1 3	3
3 1 5	3
3 1 3	8

地区比赛	
1 2 3	
3 1 3	
2 2 8	
3 1 3	
3 1 2	

地区比赛

## 问题 C. 抓住星星

包包买了一个望远镜，想观察夜空中的一颗星星。这颗星星被表示成一个凸多边形  $S$ 。然而，有  $n$  个凸多边形卫星  $M_i$  可能会阻碍他的视线。小宝可以望远镜放在  $x$  轴上介于  $(l, 0)$  和  $(r, 0)$  两点之间的任何位置，但他**不能**把望远镜准确地放在  $(l, 0)$  或  $(r, 0)$  处。

你的任务是帮助小宝找出在  $x$  轴上他可以将望远镜放置在哪个位置，从而**可以**遮挡地看到恒星  $S$  的线段总长度，以及他是否**可以**找到一个位置。视线无遮挡指的是从  $x$  轴上所选点到  $S$  内或  $S$  上任何一点的线段都不与任何卫星  $M_i$  相交。线段可以触及卫星的边界，但不能穿过它们。

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 2.5 \times 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行包含三个整数  $n$ 、 $l$  和  $r$  ( $1 \leq n \leq 10^4$ ,  $-10^9 \leq l < r \leq 10^9$ )，表示卫星的数量和望远镜的放置范围。

第二行描述恒星  $S$ ，以整数  $k_0$  开始 ( $3 \leq k_0 \leq 10^5$ )，表示数的顶点，然后是  $2 \times k_0$  个整数  $x_{(0)(1)}, y_{(0)(1)}, x_{(0)(2)}, y_{(0)(2)}, \dots, x_{(0)(k_0)}, y_{(0)(k_0)}$  ( $-10^9 \leq x_{(0)(j)}, y_{(0)(j)} \leq 10^9$ )，其中， $(x_{0,j}, y_{0,j})$  是  $S$  的第  $j$  个顶点的坐标，按逆时针顺序排列。

在接下来的  $n$  行中，第  $i$  行描述月  $M_i$ 。每行开头都有一个整数  $k_{(i)}$  ( $3 \leq k_{(i)} \leq 10^5$ )，表示  $M_{(i)}$  的顶点数，接着是  $2 \times k_{(i)}$  整数  $x_{(i)(1)}, y_{(i)(1)}, x_{(i)(2)}, y_{(i)(2)}, \dots, x_{(i)(k_{(i)})}, y_{(i)(k_{(i)})}$  ( $-10^9 \leq x_{(i)(j)}, y_{(i)(j)} \leq 10^9$ )，其中  $(x_{(i)(j)}, y_{(i)(j)})$  是按逆时针顺序排列的  $M_{(i)}$  第  $j$  个顶点的坐标。

保证  $S$  和  $M_i$  是凸多边形。恒星  $S$ 、卫星  $M_i$  以及从  $(l, 0)$  到  $(r, 0)$  的线段不会接触或相交。但是，不同的卫星  $M_i$  可以。同一多边形的三个连续顶点不相交。

可以保证  $\sum_{i=0}^n k_i$  不超过  $10^6$ 。还保证所有测试用例的多边形总数 (即  $\sum_{i=0}^n (n+1)$ ) 超过  $5 \times 10^4$ 。

### 输出

对于每个测试用例，输出一行，其中包含  $x$  轴上有效线段的总长度，严格位于  $(l, 0)$  和  $(r, 0)$  之间，即小宝可以放置望远镜无障碍地看到  $S$  的位置。如果在  $(l, 0)$  和  $(r, 0)$  之间没有有效点，则输出 1. -

如果相对误差或绝对误差不超过  $10^{-9}$ ，则答案为正确答案。

地区比赛

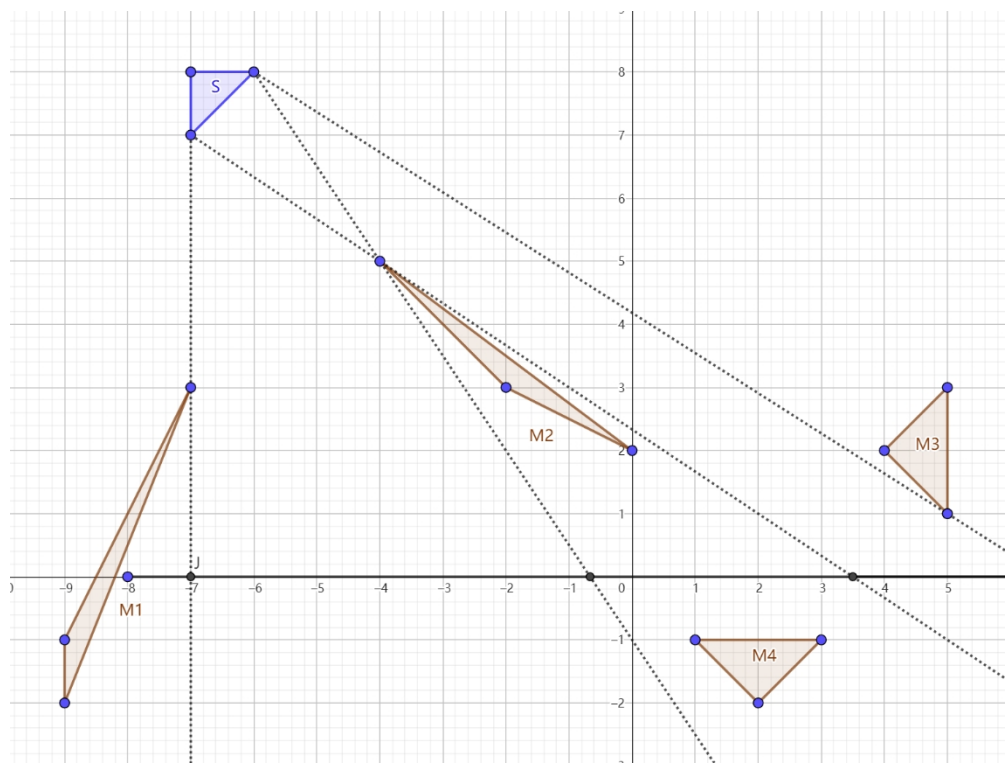
实例

标准输入	标准输出
2 4 -8 8 3 -7 7 -6 8 -7 8 3 -9 -2 -7 3 -9 -1 3 -2 3 0 2 -4 5 3 5 1 5 3 4 2 3 1 -1 2 -2 3 -1 5 -8 8 5 -14 -3 -10 -2 -9    2 -10 4 -12 5 3 -16 0 -15 0 -15 1 3 -15 6 -9 5 -15 7 3 -10 5 -9 5 -10 6 3 -7 3 -3 2 -8 4 3 -6 -1 -6 -2 -5 -1	9.404761904761905 6.000000000000000
3 1 -4 4 3 -2 6 0 5 2 6 3 -3 1 3 1 0 4 3 -2 2 3 -2 4 2 4 0 6 3 -2 2 -1 2 -2 3 3 1 2 2 2 2 3 3 -2 -1 0 -3 2 -1 1 1 2 3 -8 0 -7 0 -8 1 3 -5 0 -4 -1 -4 0	-1.000000000000000 0.000000000000000 1.000000000000000
1 1 -744567334 955216804 5 -781518205 -852078097 -781516900 -852078384 -781516392 -852076569 -781518329 -852076047 -781519925 -852077600 5 -393011614 -131855702 -393010699 -131856607 -393008846 -131856475 -393009388 -131854587 -393010201 -131854694	1699779738.691979192313738

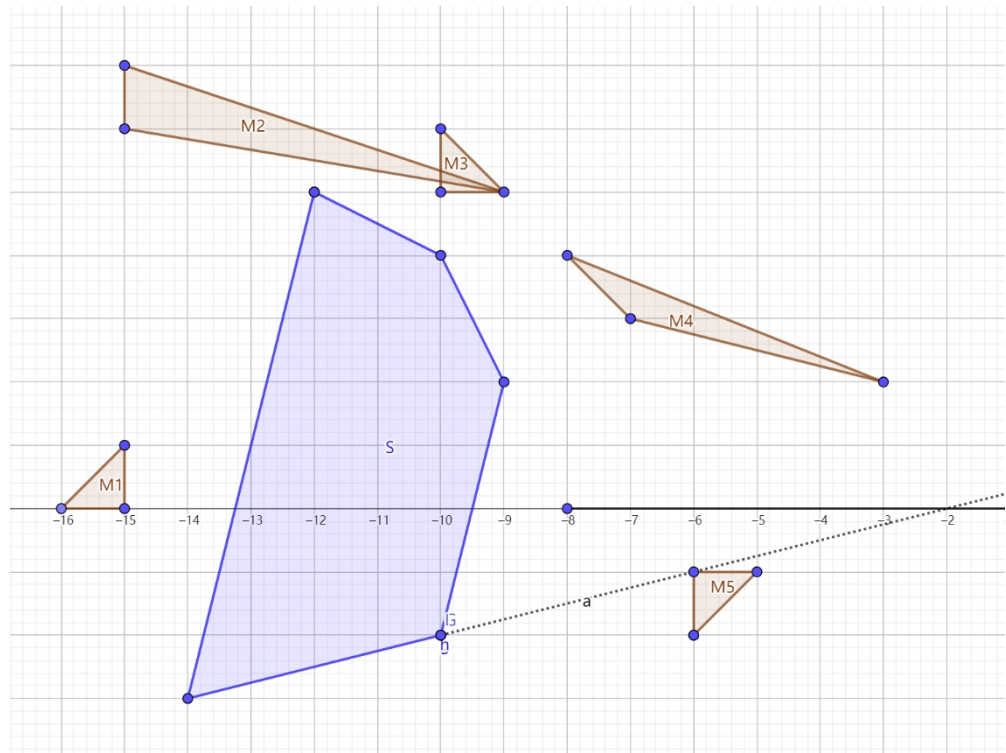
备注

对于第一个样本，第一个测试案例图所示；望远镜可以位于  $(-7, 0)$  之间。  
和  $(-\frac{2}{3}, 0)$ ，或  $(\frac{7}{2}, 0)$  和  $(\frac{46}{7}, )$ 。

地区比赛



第二个测试案例如下图所示；望远镜可以位于  $(-8, 0)$  和  $(-2, 0)$  之间。



请注意，第三个示例中的输入格式仅供演示之用，因为我们无法在不超出纸张宽度的情况下在一行中打印所有坐标。有关更精确的输入格式，请参阅其他示例。

地区比赛

问题 D. 除法序列

爱丽丝得到了一个由她的邻居构建的序列  $A$ 。由于爱丽丝不喜欢长序列，她决定把序列分成两个（可能是空的）序列  $B$  和  $C$ ，然后还给她的邻居。她的分割应满足以下限制条件：

- $B$  和  $C$  都是序列  $A$  的子序列。
- $A$  中的每个元素正好属于  $B$  或  $C$  中的一个序列。
- $B \leq C$  按词典顺序排列。

在此，我们定义长度为  $u$  的序列  $P = p_1, p_2, \dots, p_u$  在词法上小于序列  $Q = q_1, q_2, \dots, q_v$ ，长度为  $v$  的  $q_v$ ，如果以下约束条件之一为真：

- $u < v$ ， $P$  是  $Q$  的前缀。
- 存在一个整数  $1 \leq k \leq \min(u, v)$ ，使得对于所有  $1 \leq i < k$ ， $p_i = q_i$ ， $p_k < q_k$ 。

作为一个公平的女孩，爱丽丝希望公平地进行除法使  $C$  的秩尽可能小。请告诉爱丽丝  $C$  的最小值。

输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行包含一个整数  $n$  ( $1 \leq n \leq 5 \times 10^3$ )，表示序列  $A$  的长度。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^5$ )，其中  $a_i$  是序列的第  $i$  个元素  $A$ 。

保证所有测试用例的  $n$  之和不超过  $10^4$ 。

输出

对每个测试用例输出两行。然后输出第二行，包含  $m$  个整数  $c_1, c_2, \dots, c_m$ ，中间用空格隔开，其中  $c_i$  是最优  $C$  的第  $i$  个元素。

示例

标准输入	标准输出
5	1
5	3
3 1 2 3 2	3
3	1 1 2
1 1 2	2
3	3 3
3 3 3	3
5	1 3 1



地区比赛 <sup>1</sup> 5 2 2 1 3 3	4 2 1 3 3
-------------------------------------	--------------

地区比赛

问题 E. 电梯 II

有一栋楼有  $10^9$  层，但只有一部电梯。起初，电梯在  $f$ 。

有  $n$  个人在等电梯。第  $i$  人目前在第  $l_i$  层，想乘电梯到第  $r_i$  层 ( $l_i < r_i$ )。由于电梯很小，每次最多只能运载 1。

电梯上行 1 层需要消耗 1 单位电能。如果电梯向下运行，则不需要电能。也就是说，将电梯从  $x$  层移到  $y$  层需要消耗  $\max(y - x, 0)$  个单位的电能。

找出将所有人送往目的地的最佳顺序，使总电能成本最小。

更正式地说，设  $a_1, a_2, \dots, a_n$  为  $n$  的排列组合，其中  $a_{(i)}$  表示第  $i$  个乘坐电梯的人是  $a_{(i)}$ 。总电能成本可计算为

$$\sum_{i=1}^n (\max(l_{a_i} - r_{a_{(i-1)}}, 0) + r_{a_{(i)}} - l_{a_{(i)}})$$

其中，为方便起见， $a_0 = 0, r_0 = f$ 。

回想一下，长度为  $n$  的序列  $a_1, a_2, \dots, a_n$  是  $n$  的置换，当且仅当从 1 开始的每个整数都是  $n$  的置换。

到  $n$ （包括  $n$ ）的数列中正好出现一次。

输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $f$  ( $1 \leq n \leq 10^5, 1 \leq f \leq 10^9$ )，表示电梯的人数和初始位置。

在下面的  $n$  行中，第  $i$  行包含两个整数  $l_i$  和  $r_i$  ( $1 \leq l_i < r_i \leq 10^9$ )，表示第  $i$  人想乘电梯从第  $l_i$  层到第  $r_i$  层。

保证所有测试用例的  $n$  之和不超过  $3 \times 10^5$ 。

输出

对于每个测试案例，首先输出一行，其中包含一个整数，表示最小总电能然后输出另一行，其中包含  $n$  个整数  $a_1, a_2, \dots, a_n$ ，中间用空格隔开，表示搬运所有人的最佳顺序。请注意，这  $n$  个整数必须是  $n$  的排列组合。

示例

标准输入	标准输出
2	11
4 2	2 1 4 3

地区比赛	5	
1 3	2 1	
2 7		
5 6		
2 5		
2 4		
6 8		

地区比赛

问题 F. 模糊排序

在 Pigeland，有  $n$  所大学，编号从 1 到  $n$ 。每年，一些排名机构都会公布这些大学的排名。今年有  $k$  个排名榜，每个榜单都是由 1 到  $n$  个整数组成的排列组合，代表各所大学。在每次排名中，一所大学越接近排列的开头，它在该列表中的排名就越好。

QS	Zhejiang University (#42)	>	Washington University in St. Louis (#118)
ARWU	Washington University in St. Louis (#23)	>	University of Michigan - Ann Arbor (#26)
QS	University of Michigan - Ann Arbor (#25)	>	University of Toronto (#34)
Times	University of Toronto (#18)	>	Cornell University (#22)
ARWU	Cornell University (#12)	>	University of Pennsylvania (#15)
QS	University of Pennsylvania (#13)	>	Princeton University (#16)
ARWU	Princeton University (#6)	>	California Institute of Technology - Caltech (#9)
Times	California Institute of Technology - Caltech (#2)	>	Massachusetts Institute of Technology - MIT (#5)
Therefore, Zhejiang University is better than Massachusetts Institute of Technology - MIT.			

2024 年 ICPC 世界总决赛中的真实故事。

Supigar 是一名四年级学生，他想申请 Pigeland 的博士课程，他有一套自己的方法来综合评估  $n$  所大学。他认为，当且仅当符合以下条件时，大学  $x$  优于另一所大学  $y$ ：

- $x$  在至少一个列表中的排名优于  $y$ ，或
- $x$  在至少一个列表中的排名优于  $z$  ( $z \neq x, z \neq y$ )，且  $z$  优于  $y$ 。

显然，根据这一定义，可能存在某些大学  $x$  和  $y$  成对 ( $x < y$ )，这样  $x$  优于  $y$ ，而  $y$  也优于  $x$ 。Supigar 将这样的大学对称为模糊大学对。

Supigar 有  $q$  个查询，其中第  $i$  个查询可以用三个整数  $id_i$ 、 $l_i$  和  $r_i$  表示 ( $l_i \leq r_i$ )。对于每个查询，他将考虑第  $id_i$ -th 个排名列表以及该列表中第  $l_i$ -th 个位置和第  $r_i$ -th 个位置之间（包括这两个位置）的所有大学。他想知道在这些大学中，有多少对大学是模糊的。请注意，定义模糊大学对需要考虑所有  $k$  个排名表之间的上位关系。

输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 2 \times 10^5$ )，表示测试用例的数量。对于每个测试用例

地区比赛包含三个整数  $n$ 、 $k$  和  $q$  ( $1 \leq n, k, q \leq 2 \times 10^5$ ,  $1 \leq n \times k \leq 2 \times 10^5$ )，分别表示大学、排名表和查询的数量。

对于下面的  $k$  行，第  $i$  行包含  $n$  个不同的整数  $a_{(i),(1)}$ 、 $a_{(i),(2)}$ 、 $\dots$ 、 $a_{i,n}$  ( $1 \leq a_{i,j} \leq n$ ) 表示第  $i$  个等级列表。  
对于以下  $q$  行，第  $i$  行包含三个整数  $id_i^r$ 、 $l_i^{(r)}$  和  $r_i^r$  ( $0 \leq id_i^{(r)} < k$ ,  $0 \leq l_i^{(r)}, r_i^r < n$ )  
表示排名列表的编码索引和 第  $i$  次查询的查询范围。

地区比赛

- $id_i$  的实数值等于  $((id_i^r + v_{i-1}) \bmod k) + 1$ 。
- $l_i$  的实数值等于  $((l_i^r + v_{i-1}) \bmod n) + 1$ 。
- $r_i$  的实数值等于  $((r_i^r + v_{i-1}) \bmod n) + 1$ 。

其中,  $v_{i-1}$  是  $(i-1)$  - 次查询的答案。具体来说, 我们定义  $v_0 = 0$ 。使用编码查询时, 您必须在处理下查询之前计算出每个查询的答案。在解码后, 保证  $1 \leq id_i \leq k$  和  $1 \leq l_i \leq r_i \leq n$ 。

此外, 还保证所有测试案例的  $n \times k$  和  $q$  之和都不会超过  $2 \times 10^5$ 。

输出

每个测试用例输出  $q$  行。每行包含一个整数, 代表作为第  $i$  个查询答案的模糊对的数量。

示例

标准输入	标准输出
2	3
5 2 2	10
1 2 3 4 5	1
5 4 3 2 1	1
1 0 2	2
1 2 1	
5 3 3	
1 2 3 4 5	
1 3 2 4 5	
1 2 3 5 4	
0 0 2	
0 2 3	
1 0 3	

备注

在第一个测试案例中, 两个解码查询分别是 2 1 3 和 1 1 5。

在第二个测试案例中, 三个解码查询分别是 1 1 3、2 4 5 和 3 2 5。

地区比赛

问题 G. 采集蘑菇

小宝正在森林里采蘑菇。森林里有  $n$  个地点，在第  $i$  个地点生长着无数  $t_i$  类型的蘑菇。每个地点都有一个牌。第  $i$  个地点的木牌指向地点  $a_i$ （有可能  $a_i = i$ ）。

由于森林里雾很大，为了安全起见，小宝决定根据路标在不同地点之间移动。小宝带着空篮子从地点  $s$  开始，每次走进一个地点  $c$ （包括起始地点  $c = s$ ，也不管他之前是否去过地点  $c$ ），他都会在篮子里采摘一朵类型为  $t_c$  的蘑菇，然后移动到地点  $a_{(c)}$ 。

给定一个整数  $k$ ，对于每个  $1 \leq s \leq n$ ，确定至少出现  $k$  种蘑菇的第一种类型次放入篮筐。

输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $k$  ( $1 \leq n \leq 2 \times 10^5$ ,  $1 \leq k \leq 10^9$ )，表示蘑菇出现的地点数量和所需时间。

第二行包含  $n$  个整数  $t_1, t_2, \dots, t_n$  ( $1 \leq t_i \leq n$ )，其中  $t_i$  是第  $i$  个地点生长的蘑菇类型。

第三行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ )，其中  $a_i$  是位置  $i$  中的符号所指向的位置。

保证所有测试用例的  $n$  之和不超过  $2 \times 10^5$ 。

输出

为减少输出的大小，每个测试用例输出一行，每行包含一个整数，表示  $\sum_{i=1}^n (i \times v_i)$ ，其中  $v_i$  是  $s = i$  的答案。

示例

标准输入	标准输出
3	41
5 3	45
2 2 1 3 3	14
2 5 1 2 4	
5 4	
2 2 1 3 3	
2 5 1 2 4	
3 10	
1 2 3	
1 3 2	

备注

对于第一个示例， $v_1 = 2$ ,  $v_2 = 3$ ,  $v_3 = 2$ ,  $v_4 = 3$ ,  $v_5 = 3$ ，因此应  $1 \times 2 + 2 \times 3 + 3 \times 2 + 4 \times 3 + 5 \times 3 = 41$ 。考虑  $s = 3$ ，包包挑选的蘑菇类型依次为  $\{1, 2, 2, 3, 3, 2, \dots\}$ ，因此类型 2 的蘑菇是第一个在篮子里至少出

地区比赛类型。



地区比赛

问题 H. 重-轻分解

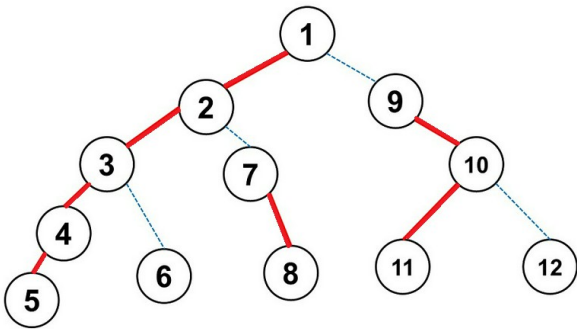
重轻分解（HLD）是一种适用于树的有用技术，可用于高效查询顶点链。让我们先回顾一下 HLD 的定义，以防忘记。

给你一棵有根的树，树上有  $n$  个顶点，编号从 1 到  $n$ 。每个非叶顶点正好有一个子顶点被划分为重顶点，其余顶点被划分为轻顶点。

对于任何非根顶点  $v$ ，让  $u$  成为它的父顶点。只有当顶点  $v$  的子树的大小是  $u$  的所有子树中最大的一个时，它才能被归类为重型顶点  $v$ 。更正式地说，将根植于顶点  $v$  的子树的大小记为  $s_v$ ，让  $\text{ch}(u)$  表示  $u$  的子树集。那么，只有当所有  $w \in \text{ch}(u)$  的  $s_v \geq s_w$  都是重型顶点  $v$  时，它才是重型顶点  $v$ 。请注意，可能有多个  $u$  的子顶点满足这一约束条件；在这种情况下，应该选择其中一个为重顶点，其他的为轻顶点。

，树的所有顶点都可以分解成若干个不重叠的重链，其中每个顶点都正好属于一个重链。重链是顶点  $x_1$ 、 $x_2$ 、.....、 $x_{(k)}$  的序列。  $x_k$  满足以下所有约束条件。

- $x_1$  要么是根顶点，要么是光顶点。
- 对于所有  $2 \leq i \leq k$ ， $x_i$  是  $x_{(i)-1}$  的子顶点，是一个重顶点。
- $x_k$  是一片叶子。



HLD 示例。重链用红色实边标记。

例如，上图显示了一棵有 12 个顶点的树的有效 HLD，其中重链为 [1、2、3、4、5]、[9、10、11]、[7、8]、[6] 和 [12]。

Pig100Ton 是 Pigeland 的一名经验丰富的编程竞赛选手。他想知道是否有可能从 HLD 后的重链中恢复出原始树。具体来说，他会给出原始树的顶点数  $n$  和  $k$  个重链。第  $i$  个链由两个整数  $l_i$  和  $r_i$  描述，表示重链  $l_{(i)}$ ， $l_i + 1, \dots, r_{i\circ}$

你的任务是构建一棵满足以下约束条件的树，或者告诉 Pig100Ton 这是不可能的：

- 它是一棵有根的树，包含从 1 到  $n$  编号的  $n$  个顶点。

地区比赛 100Ton 提供的  $k$  个链应构成一棵树的有效 HLD。有效的 HLD 是指将每个非根顶点划分为轻链或重链，然后将树分解多个不重叠的重链的有效方法。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^5$ )，表示测试用例的数量。对于每个测试用例

地区比赛  
第一行包含两个整数  $n$  和  $k$  ( $1 \leq n \leq 10^5, 1 \leq k \leq n$ )，表示树的顶点数和 HLD 后的重链数。

对于下面的  $k$  行，第  $i$  行包含两个整数  $l_i$  和  $r_{(i)}$  ( $1 \leq l_i \leq r_{(i)} \leq n$ )，表示第  $i$  个重链为  $l_i, l_i + 1, \dots, r_{(i)}$ 。

保证  $n$  个顶点中的每个顶点都正好属于一个给定的重链。同时保证所有测试案例的  $n$  之和不超过  $2 \times 10^5$ 。

输出

对于每个测试用例

如果可以构建这样一棵树，请输出一行包含  $n$  个整数  $p_1, p_2, \dots, p_n$ ，中间用空格隔开，其中  $p_i$  是顶点  $i$  的父节点。 $p_i = 0$  表示顶点  $i$  是根节点。如果有多个有效解，您可以输出其中任意一个。

如果无法构建这样一棵树，只需在一行中输出 IMPOSSIBLE。

示例

标准输入	标准输出
3	0 1 2 3 4 3 2 7 1 9 10 10
12 5	2 0 2 2
1 5	不可能
9 11	
7 8	
6 6	
12 12	
4 3	
1 1	
4 4	
2 3	
2 2	
1 1	
2 2	

备注

第一个测试用例的样本输出是描述中显示的树形。

地区比赛

## 问题 1. 确定和弦

这是一个互动问题。

Grammy 有一个由  $n$  ( $4 \leq n \leq 10^9$ ) 个从 1 到  $n$  的顶点组成的无向循环图。无向循环图是由  $n$  个顶点和  $n$  条无向边组成一个循环的图。具体来说，顶点  $i$  和顶点  $((i \bmod n) + 1)$  之间有一条双向边，每  $1 \leq i \leq n$ 。

格莱美认为这个图太无聊了，于是她偷偷选择了一对不相邻的顶点，并在它们之间连接了一条无向边（称为弦），这样这个图就包含了  $n$  个顶点和  $(n + 1)$  条边。

你的任务是通过不超过 40 次的查询来猜测弦的位置。每个查询包括两个顶点  $x$  和  $y$ ，Grammy 会告诉你这两个顶点之间最短路径上的边的数量。

请注意，交互器是非适应性的，也就是说，和弦的位置是预先确定的。

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^3$ )，表示测试用例的数量。对于每个测试用例

第一行包含一个整数  $n$  ( $4 \leq n \leq 10^9$ )，表示顶点数。

### 互动协议

要进行查询，只需输出一行。首先输出 "?"，后面跟一个空格，然后输出两个顶点  $x$  和  $y$  ( $1 \leq x, y \leq n$ )，中间用空格隔开。刷新输出后，程序应读取一个整数，表示两个顶点之间最短路径上的边数。

要猜测弦的位置，只需输出一行即可。首先输出 "!" 之后的空格，然后输出两个顶点  $u$  和  $v$  ( $1 \leq u, v \leq n$ )，中间用空格隔开，表示弦连接顶点  $u$  和  $v$ 。

刷新输出后，程序应读取一个整数  $r$  ( $r \in \{1, -1\}$ )，表示猜测的正确性。如果  $r = 1$ ，则表示猜测正确，程序应继续处理下一个测试用例，如果没有测试用例，则立即退出。否则，如果  $r = -1$ ，则您的猜测不正确，您的程序应立即退出，并收到 "错误答案" 判决。请注意，您的猜测不算是一次查询。

要冲洗输出，可以使用

- 在 C 和 C++ 中，`fflush(stdout)`（如果使用 `printf`）或 `cout.flush()`（如果使用 `cout`）。
- Java 中的 `System.out.flush()`。
- 在 Python 中使用 `stdout.flush()`。

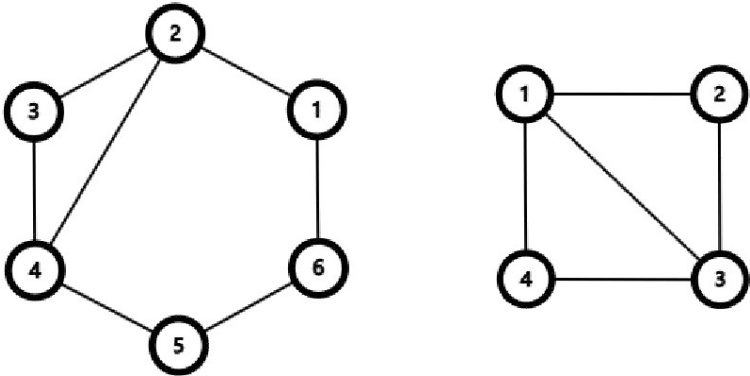
地区比赛

示例

标准输入	标准输出
2	
6	? 1 5
2	? 2 4
1	! 4 2
1	
4	? 2 4
2	! 1 3
1	

备注

示例测试用例中的图表说明如下：



地区比赛

问题 J. 日本乐队

格莱美正在设计一款新的交易卡游戏（TCG），该游戏以她最喜欢的日本音乐媒体系列 *BanG Dream!* 按照她的设计，一共有  $n_1$  张人物卡和  $n_{(2)}$  张音乐卡。现在，她需要为每张卡分配一个介于 1 和  $m$  之间的整数（都包括在内），代表它所包含的魔力。

在每个 TCG 游戏中，一定有一些组合可能会造成额外的伤害。格莱美现在考虑的是一种与牌的赋值有关的特殊规则。具体地说，要选择  $k$  对整数  $(a_{(1)}, b_{(1)}), (a_2, b_2), \dots, (a_k, b_{(k)})$ ，满足  $1 \leq a_i, b_i \leq m$ 。格莱美希望确保这些数值组合都能在游戏中使用。因此，对于每一对整数  $(a_i, b_i)$ ，其分配必须至少满足以下两个限制条件中的一个：

- $a_i$  可以在人物卡上找到， $b_{(i)}$  可以在音乐卡上找到。
- $a_i$  可以在音乐卡上找到， $b_{(i)}$  可以在人物卡上找到。请帮助格莱美计算有效卡值分配的数量。

假设  $C$  是字符卡上整数的多重集合， $M$  音乐卡上整数的多重集合。如果两个赋值的  $C$  不同或  $M$  不同，我们就说这两个赋值是不同的。

回顾一下，一个整数可以在多重集合中出现多次。如果存在一个整数  $k$ ，使得  $k$  在  $X$  中出现的次数与在  $Y$  中出现的次数不相等，我们就说两个多重集合  $X$  和  $Y$  是不同的。

输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 500$ )，表示测试用例的数量。对于每个测试用例

第一行包含四个整数  $n_1$ 、 $n_2$ 、 $m$  和  $k$  ( $1 \leq n_1, n_{(2)} \leq 10^9$ 、 $1 \leq m \leq 20$ 、 $1 \leq k \leq m^2$ )。对于下

面的  $k$  行，第  $i$  行包含两个整数  $a_i$  和  $b_i$  ( $1 \leq a_i, b_{(i)} \leq m$ )。

保证最多有 5 个测试用例满足  $m > 10$ 。

输出

对每个测试用例输出一行，其中包含一个整数，表示有效的卡值分配数。由于答案可能很大，请输出答案的模数  $(10^9 + 7)$ 。

示例

标准输入	标准输出
3	6
2 3 3 3	4
2 3	0
1 1	
2 3	
2 2 2 1	

地区比赛 1 1 10 2 1 2 1 3	
--------------------------------	--

备注

对于第一个示例，（C，M）的有效配对是（{1，2}，{1，1，3}）、（{1，2}，{1，2，3}）、（{1，2}，{1，3，3}）、（{1，3}，{1，1，2}）、（{1，3}，{1，2，2}）和（{1，3}，{1，2，3}）。

对于第二个样本测试用例，（C，M）的有效配对是（{1，1}，{1，1}）、（{1，2}，{1，1}）、（{1，1}，{1，2}）。

和（{1，2}，{1，2}）。

地区比赛

# 问题 K. 一种宾果

有一个  $n$  行  $m$  列的网格。网格中的单元格编号从 1 到  $n \times m$ ，其中第  $i$  行第  $j$  列上的单元格编号为  $((i-1) \times m + j)$ 。

给定  $n \times m$  的排列  $p_1, p_2, \dots, p_{(n) \times m}$ ，我们将根据排列执行  $n \times m$  次操作。对于第  $i$  次操作，我们将标记单元格  $p_i$ 。如果在第  $b$  次操作后，至少有一行中的所有单元格都被标记，并且  $b$  越小越好，那么我们就说  $b$  是该排列组合的 "宾果整数"。

你最多有  $k$  次机会修改排列组合（包括次）。每次你都可以交换排列中的一对元素。计算修改后最小的宾果整数。

回想一下，长度为  $n \times m$  的序列  $p_1, p_2, \dots, p_{(n) \times m}$  是  $n \times m$  的置换，当且仅当从 1 到  $n \times m$ （包括这两个整数）的每个整数在序列中恰好出现一次。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行包含三个整数  $n$ 、 $m$  和  $k$  ( $1 \leq n, m \leq 10^5$ ,  $1 \leq n \times m \leq 10^5$ ,  $0 \leq k \leq 10^9$ )，表示网格的行数和列数以及可以执行的修改次数。

第二行包含  $n \times m$  个不同的整数  $p_1, p_2, \dots, p_{n \times m}$  ( $1 \leq p_i \leq n \times m$ )。保证所有测试用例的  $n \times m$  之和不超过  $10^5$ 。

## 输出

对每个测试用例输出一行，其中包含一个整数，表示修改后可能的最小宾果整数。

## 示例

标准输入	标准输出
3	7
3 5 2	5
1 4 13 6 8 11 14 2 7 10 3 15 9 5 12	3
2 3 0	
1 6 4 3 5 2	
2 3 1000000000	
1 2 3 4 5 6	

## 备注

对于第一个示例，我们可以先交换 1 和 15，然后交换 6 和 12，得到序列 [15, 4, 13, 12, 8, 11, 14, 2, 7, 10, 3, 1, 9, 5, 6]。不难看出，经过第 7 次操作后，第 3 行的所有单元格都将被标记。



对于第二个示例测试用例，我们不难发现，在第 5 次操作后，第 2 行的所有单元格都会被标记。

对于第三个示例，我们不需要做任何修改。不难看出，在第 3 次操作时，第 1 行的所有单元格都将被标记。

地区比赛

## 问题 L.我们出发吧--新的冒险

在《猪猪侠》中，*Pishin* 是一款广受欢迎的开放世界动作 RPG 游戏，用户可以扮演多个角色。每个角色都有一个独立的 *冒险等级*，该等级会随着他们在游戏中获得的经验值（EXP）而增加。初始时，每个角色的冒险等级为 0 级，最高可升至  $m$  级。要从等级  $(i-1)$  升至  $i$  级  $(1 \leq i \leq m)$ ，角色需要获得  $b_i$  点经验值。当前等级越高，升级就越困难，也就是说， $b_i \leq b_{(i+1)}$  对从 1 到  $m$  的所有  $i$  始终成立。

格莱美计划在接下来的  $N$  天里玩 *Pishin*。作为一个富家女，她的 *Pishin* 账户拥有无限多的角色。然而，作为一个懒惰的女孩，她账户中的所有角色在  $n$  开始时的冒险等级都是 0 级。，格莱美都会选择一个角色进行游戏，但一旦她停止玩某个角色，就无法在以后的任何一天继续玩该角色。换句话说，她只能在连续的日子里继续扮演同一个角色。

在第  $i$  天，格莱美将为她扮演的角色获得  $a_i$  EXP。这意味着，如果她从第  $l$  天到第  $r$  天（包括第  $r$  天）连续扮演一个角色，该角色的冒险等级就会提高到  $k$  级，其中  $k$  是 0 到  $m$  之间的最大整数，使得赚取的总 EXP

(即 $\sum_{i=l}^ra_i$ ) 大于或等于平移到  $k$  的要求。 $(即\sum_{i=1}^kb_i)$  。

Grammy 是个贪心的女孩，她希望在  $N$  天后最大限度地提高所有角色的冒险等级总和。然而，作为一个一心一意的女孩，她不想扮演太多不同的角色。为了平衡这一点，她引入了一个惩罚因子  $c$ 。她的目标是在  $n$  天后最大化所有角色的冒险等级总和，减去  $c \times d$ ，其中  $d$  是她扮演的不同角色的数量。作为格莱美最好的朋友，你的任务是计算出她在最佳角色选择策略下能达到最大值。

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$   $(1 \leq T \leq 5 \times 10^4)$ ，表示测试用例的数量。对于每个测试用例

第一行包含三个整数  $n$ 、 $m$  和  $c$   $(1 \leq n, m \leq 5 \times 10^5, 0 \leq c \leq 5 \times 10^{(5)})$ 。

第二行包含 $n$  个整数  $a_1, a_2, \dots, a_n$   $(0 \leq A_i \leq 10^{12}, \leq \sum_{i=1}^na_i \leq 10^{(12)})$ 。

第三行包含 $m$  个整数  $b_1, b_2, \dots$   $(0 \leq b_i \leq 10^{12}、\ b_i \leq b_{i+1} \ \leq \sum_{i=1}^mb_i \leq 10^{(12)})$ 。  
， $b_m$

保证所有测试案例的  $n$  和  $m$  之都不会超过  $5 \times 10^5$ 。

### 输出

对于每个测试用例，输出一行包含一个整数的数据，表示最大值。

### 示例

标准输入	标准输出
------	------

地区比赛	3
5 4 2	6
1 0 3 1 2	
0 1 1 2	
4 5 1	
7 16 23 4	
1 3 6 20 20	

备注

对于第一个示例，一种解决方案是利用前三天的时间获得一个具有冒险精神的角色

地区比赛

等级 4，接下来的两天再获得一个冒险等级 3 的角色。这样我们就可以得到  $(4-2)+(3-2)=3$ 。

在第二个测试案例中，我们可以每天扮演不同的角色，这样就能获得冒险等级分别为 2、3、3 和 2。因此值为  $(2-1)+(3-1)+(3-1)+(2-1)=6$ 。

地区比赛

# 问题 M. 可分割

给定长度为  $n$  的序列  $a_1, a_2, \dots, a_n$  包含正整数，如果存在整数  $d$ ，使得  $l \leq d \leq r$ ，并且对于所有  $l \leq i \leq r$ ， $a_i$  都能被  $a_d$  整除，我们就说区间  $[l, r]$  ( $1 \leq l \leq r \leq n$ ) 是可整除区间。如果对于所有  $1 \leq l \leq r \leq n$ ， $[l, r]$  都是可分区间，我们就说整个序列是可分序列。

给定长度为  $n$  的另一个序列  $b_1, b_2, \dots, b_n$  和一个整数  $k$ ，求所有整数  $x$ ，使得  $1 \leq x \leq k$  和序列  $b_1 + x, b_2 + x, \dots, b_n + x$  是一个可除序列。由于这样的整数可能很多，因此只需输出所有这样的整数的个数和总和即可。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 500$ )，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $k$  ( $1 \leq n \leq 5 \times 10^4, 1 \leq k \leq 10^9$ )。第二行包含  $n$  个整

数  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 10^9$ )。

保证所有测试用例的  $n$  之和超过  $5 \times 10^4$ 。

## 输出

对每个测试用例输出一行，其中包含两个用空格隔开的整数，第一个整数是有效  $x$  的个数，第二个整数是所有有效  $x$  的总和。

## 示例

标准输入	标准输出
3	3 8
5 10	0 0
7 79 1 7 1	100 5050
2 1000000000	
1 2	
1 100	
1000000000	

## 备注

对于第一个样本测试用例， $x=1$ 、 $x=2$  和  $x=5$  均有效。对于

第三个示例，所有  $1 \leq x \leq 100$  都有效。