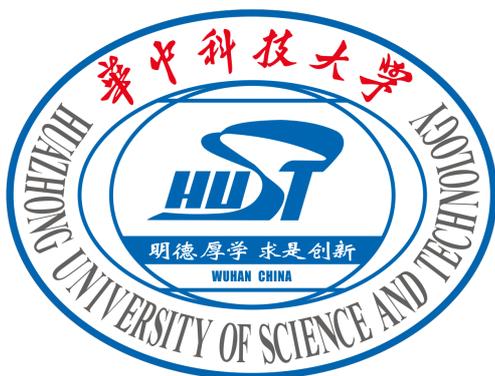


2025  icpc 国际大学生程序设计竞赛  
全国邀请赛（武汉）

正式赛

2025 年 4 月 27 日



试题列表

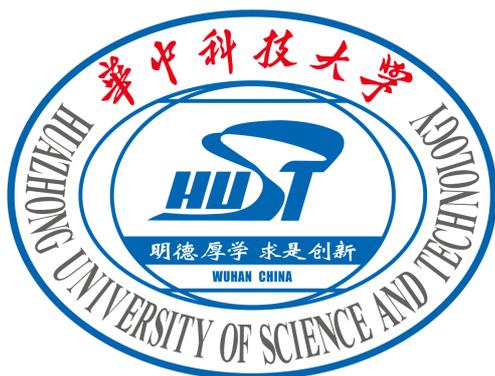
|   |                |
|---|----------------|
| A | 出题             |
| B | 黑红树            |
| C | 我们必须想象西西弗斯是幸福的 |
| D | 奇与偶            |
| E | 多彩的图           |
| F | 背包             |
| G | 路径求和问题         |
| H | 野火哥，你行不行？      |
| I | Bingo 3        |
| J | 字典             |
| K | 拉斯维加斯          |
| L | 子序列            |
| M | 航班追踪器          |

本试题册共 13 题，18 页。  
如果您的试题册缺少页面，请立即通知志愿者。

由 SUA 程序设计竞赛命题组命题。

<https://sua.ac/>

## 承办方



## 命题方



竞赛过程中访问非竞赛网页是违反竞赛规则的行为。  
如果您有兴趣（我们很荣幸），  
请在竞赛后扫描二维码。

## Problem A. 出题

SUA 程序设计竞赛命题组的成员们正在为 2025 ICPC 国际大学生程序设计竞赛全国邀请赛（武汉）准备试题。他们正在出的一道题可以从不同方面用  $n$  个属性描述，例如难度、代码长度等等。第  $i$  个属性的值为  $a_i$ 。

成员们同时提出了  $q$  条建议，第  $i$  条建议可以表示为三个整数  $p_i$ ,  $l_i$  和  $r_i$ ，表示第  $p_i$  个属性的值需要在  $l_i$  到  $r_i$  之间（含两端）。

堡堡是这道题的作者，他准备根据这些建议修改题目。他可以花一单位时间将一个属性的值增加或减少 1。求满足所有建议最少需要多少时间，或表明不可能满足所有建议。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 100$ ) 表示测试数据组数，对于每组测试数据：

第一行输入两个整数  $n$  和  $q$  ( $1 \leq n, q \leq 100$ ) 表示属性的数量以及建议的数量。

第二行输入  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ )，其中  $a_i$  是第  $i$  个属性的值。

对于接下来  $q$  行，第  $i$  行输入三个整数  $p_i$ ,  $l_i$  和  $r_i$  ( $1 \leq p_i \leq n$ ,  $1 \leq l_i \leq r_i \leq 10^9$ )，表示第  $p_i$  个属性的值需要在  $l_i$  到  $r_i$  之间（含两端）。

### Output

每组数据输出一行一个整数，表示满足所有建议最少需要多少时间。如果不可能满足所有建议，则输出 -1。

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | 6               |
| 4 3            | -1              |
| 20 25 4 27     | 0               |
| 3 5 7          |                 |
| 1 10 15        |                 |
| 3 2 6          |                 |
| 1 2            |                 |
| 7              |                 |
| 1 3 5          |                 |
| 1 9 9          |                 |
| 1 2            |                 |
| 7              |                 |
| 1 3 9          |                 |
| 1 4 15         |                 |

### Note

对于第一组样例数据，堡堡可以将第 1 个属性的值变为 15，将第 3 个属性的值变为 5。答案是  $(20 - 15) + (5 - 4) = 6$ 。

对于第三组样例数据，由于  $3 \leq 7 \leq 9$  以及  $4 \leq 7 \leq 15$ ，所有建议已经被满足了。堡堡不需要修改任何属性的值。

## Problem B. 黑红树

请注意本题不同寻常的空间限制。

堡堡有一棵  $n$  个节点的树。一开始，树中所有的边都是黑色的。因为堡堡喜欢对树进行染色和计数，所以他会进行  $(n - 1)$  次染色操作。对于第  $i$  次操作，他会将第  $q_i$  条边变成红色。显然所有操作结束后，所有边都会变成红色。

您需要计算每次操作后，树上有几条简单路径恰好包含  $k$  条黑边。请注意，我们认为从节点  $u$  到  $v$  的简单路径，和从节点  $v$  到  $u$  的简单路径是同一条路径。

请回忆：简单路径不会多次经过同一条边。

### Input

每个测试文件仅有一组测试数据。

第一行输入两个整数  $n, k$  ( $2 \leq n \leq 2 \times 10^5, 1 \leq k \leq 10$ ) 表示树中的节点数和要求的路径上的黑边数。

对于接下来的  $(n - 1)$  行，第  $i$  行输入两个整数  $u_i$  和  $v_i$  ( $1 \leq u_i, v_i \leq n$ )，表示树上的第  $i$  条边连接了节点  $u_i$  和  $v_i$ 。

对于接下来的  $(n - 1)$  行，第  $i$  行输入一个整数  $q'_i$  ( $1 \leq q'_i < n$ )，表示堡堡染色的第  $i$  条边的编号加密后的值。解密后的真实值  $q_i$  等于  $((q'_i + a_{i-1}) \bmod (n - 1)) + 1$ ，其中  $a_{i-1}$  是第  $(i - 1)$  次询问的答案。特别地，我们定义  $a_0$  为堡堡染色前树上恰好包含  $k$  条黑边的简单路径数量。由于询问是加密的，您需要依次计算每次询问的答案后，再处理下一次询问。输入数据保证  $q_1, q_2, \dots, q_{n-1}$  是  $1, 2, \dots, n - 1$  的排列。

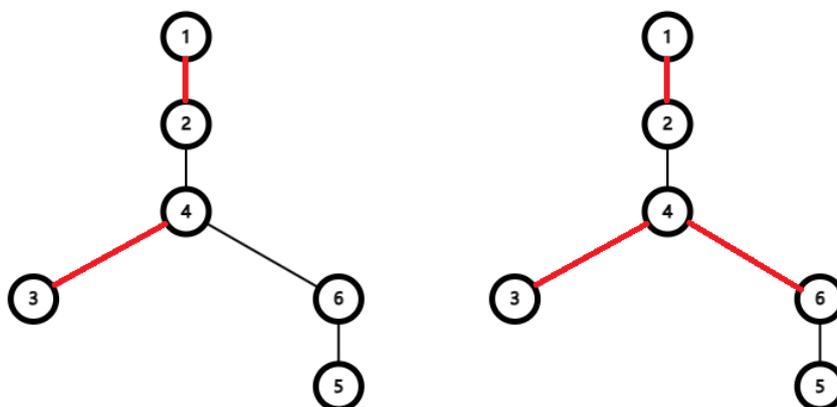
### Output

输出  $(n - 1)$  行，其中第  $i$  行输出一个整数，表示第  $i$  次染色操作后，有多少条简单路径恰好包含  $k$  条黑边。

## Examples

| standard input                                                  | standard output       |
|-----------------------------------------------------------------|-----------------------|
| 6 1<br>4 6<br>4 2<br>4 3<br>5 6<br>1 2<br>4<br>2<br>1<br>2<br>3 | 5<br>7<br>8<br>8<br>0 |
| 6 2<br>4 6<br>4 2<br>4 3<br>5 6<br>1 2<br>2<br>5<br>5<br>4<br>3 | 5<br>4<br>2<br>0<br>0 |

## Note



对于第一组样例数据， $q_1, q_2, q_3, q_4, q_5$  的真实值分别是 5, 3, 4, 1, 2。前两次操作后的树如左图所示。黑边用更细的线表示，而红边用更粗的线表示。

令  $u \rightarrow v$  表示从节点  $u$  到  $v$  的简单路径。恰好包含一条黑边的简单路径有： $1 \rightarrow 3$ ,  $1 \rightarrow 4$ ,  $2 \rightarrow 3$ ,  $2 \rightarrow 4$ ,  $3 \rightarrow 6$ ,  $4 \rightarrow 6$ ,  $5 \rightarrow 6$ 。

对于第二组样例数据， $q_1, q_2, q_3, q_4, q_5$  的真实值分别是 3, 1, 5, 2, 4。前三次操作后的树如右图所示。

恰好包含两条黑边的简单路径有： $1 \rightarrow 5$ ,  $2 \rightarrow 5$ 。

## Problem C. 我们必须想象西西弗斯是幸福的

西西弗斯被任命去开垦一片长条形的耕地。

耕地从左到右被划分成  $n$  格，编号为 1 到  $n$ 。刚开始这里长满了杂草，西西弗斯站在格子 1。他将从格子 1 走到格子  $n$ ，再走回格子 1，期间他每经过一个格子都需要 1 小时去检查并清除发现的杂草（无论这一格是否真的有杂草）。西西弗斯在两格间移动的时间忽略不计，因此一个来回他需要  $(2n - 1)$  小时，其中前  $(n - 1)$  个格子会检查两次，而第  $n$  个格子只会检查一次。来回一轮后他需要报告有多少格的杂草被清理了，报告所需的时间同样忽略不计。

然而耕地底下埋藏着无尽的杂草种子。当西西弗斯清理完第  $i$  格的杂草并离开该格子之后  $a_i$  小时 1 分钟，这一格杂草又会卷土重来。特殊地，如果杂草重新长出时，西西弗斯正好又在检查这一格，新长出的杂草也会被清除。

加缪想要知道，对于每个  $1 \leq k \leq m$ ，西西弗斯的第  $k$  次来回需要清理几格杂草。如果西西弗斯已经提前知道了这些数字，那他应当是幸福的。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 2 \times 10^3$ ) 表示测试数据组数，对于每组测试数据：

第一行输入两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 10^6$ )，表示耕地的格数和来回的轮数。

第二行输入  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^9$ )，第  $i$  格的杂草会在  $a_i$  小时 1 分钟后重新长出。

保证所有数据  $n$  与  $m$  的之和都不超过  $10^6$ 。

### Output

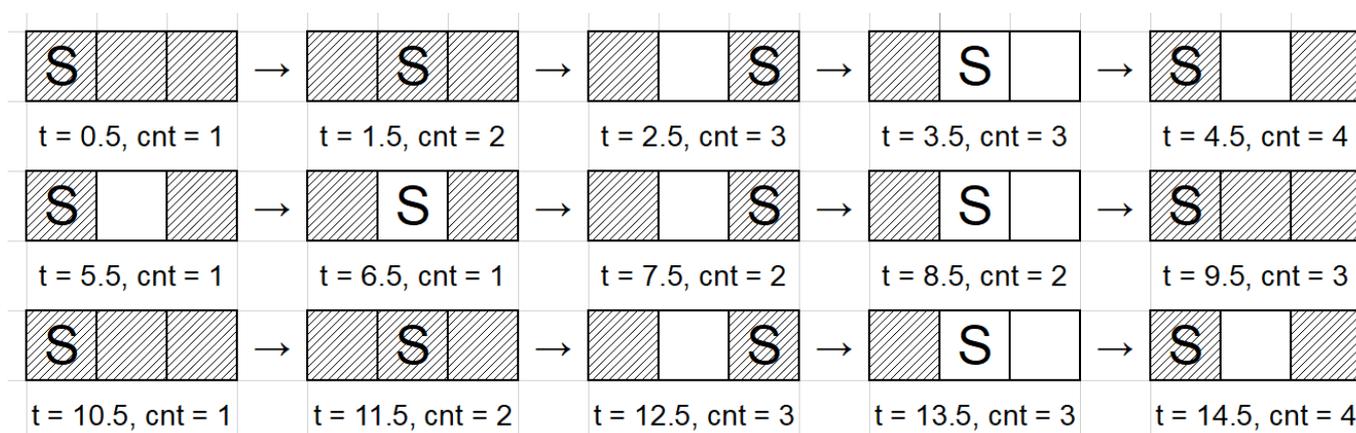
每组数据输出一行  $m$  个由单个空格分隔的整数，其中第  $i$  个数表示第  $i$  轮来回需要清理几格杂草。

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | 4 3 4           |
| 3 3            | 6 3 4 3         |
| 0 7 1          | 5 3 3           |
| 5 4            |                 |
| 6 16 10 5 10   |                 |
| 3 3            |                 |
| 2 1 20         |                 |

### Note

下图展示了第一组样例数据。空白格子表示没有杂草的耕地，阴影格子表示有杂草的耕地（包括正在被清理的杂草），S 表示西西弗斯，t 表示当前的小时数，cnt 表示这一轮已经清理了多少格杂草（包括正在被清理的杂草）。



## Problem D. 奇与偶

给定长度为  $k$  的整数序列  $A = a_1, a_2, \dots, a_k$ ，您需要将其分成若干个不为空的连续子数组，使得每个元素恰属于一个子数组。对于每个子数组，计算它的元素之和。设  $p$  表示和为奇数的子数组数量， $q$  表示和为偶数的子数组数量。

您需要回答关于这个序列的  $m$  次询问。每次询问可以表示为一个整数  $r$ ，您需要分别求出在满足  $p + q = r$  的情况下， $p$  和  $q$  可能的最大值分别是多少。

因为序列  $A$  可能很长，我们将用游程编码（run-length encoding）的方式描述该序列。更正式地，给定  $n$  对整数  $(v_1, l_1), (v_2, l_2), \dots, (v_n, l_n)$ ，序列  $A$  可以通过以下方式得出：从一个空序列开始，首先将  $v_1$  附加到序列末尾  $l_1$  次，然后将  $v_2$  附加到序列末尾  $l_2$  次，...，最后将  $v_n$  附加到序列末尾  $l_n$  次。下面对样例数据的解释可以提供例子。

### Input

每个测试文件仅有一组测试数据。

第一行输入两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 2 \times 10^5$ )，表示序列游程编码的长度，以及询问的数量。

对于接下来  $n$  行，第  $i$  行输入两个整数  $v_i$  和  $l_i$  ( $1 \leq v_i, l_i \leq 10^9$ )。因此，序列  $A$  的长度就是  $k = \sum_{i=1}^n l_i$ 。保证对于所有  $1 \leq i < n$  有  $v_i \neq v_{i+1}$ 。

对于接下来  $m$  行，第  $i$  行输入一个整数  $r_i$  ( $1 \leq r_i \leq k$ )，表示第  $i$  次询问。

### Output

每个询问输出一行两个由单个空格分隔的整数，表示在满足  $p + q = r$  的情况下， $p$  和  $q$  可能的最大值分别是多少。

## Example

| standard input | standard output |
|----------------|-----------------|
| 3 6            | 0 1             |
| 5 3            | 2 2             |
| 2 2            | 2 1             |
| 7 1            | 4 2             |
| 1              | 4 3             |
| 2              | 4 2             |
| 3              |                 |
| 4              |                 |
| 5              |                 |
| 6              |                 |

## Note

对于样例数据， $A = 5, 5, 5, 2, 2, 7$ 。

对于第二次询问，我们需要将序列分成 2 个连续子数组。

- 要最大化和为奇数的子数组数量，我们可以将  $A$  分成  $5, 5, 5 | 2, 2, 7$ 。两个子数组的和都是奇数，所以  $p$  可能的最大值是 2。
- 要最大化和为偶数的子数组数量，我们可以将  $A$  分成  $5, 5 | 5, 2, 2, 7$ 。两个子数组的和都是偶数，所以  $q$  可能的最大值是 2。

对于第五次询问，我们需要将序列分成 5 个连续子数组。

- 要最大化和为奇数的子数组数量，我们可以将  $A$  分成  $5 | 5 | 5 | 2, 2 | 7$ ，其中 4 个子数组的和是奇数，所以  $p$  可能的最大值是 4。
- 要最大化和为偶数的子数组数量，我们可以将  $A$  分成  $5, 5 | 5 | 2 | 2 | 7$ ，其中 3 个子数组的和是偶数，所以  $q$  可能的最大值是 3。

## Problem E. 多彩的图

给定一张  $n$  个节点  $m$  条边的无向图，您需要给每条边涂上  $m$  种颜色中的一种，使得对于任意一个节点  $u$  以及任意一种颜色  $c$ ，最多只有 2 条与节点  $u$  相连的边被涂上了颜色  $c$ 。

更正式地，对于所有整数  $1 \leq u \leq n$ ，都需要满足以下条件：令  $d_u$  表示与节点  $u$  相连的边的数量，令  $e_{u,1}, e_{u,2}, \dots, e_{u,d_u}$  表示这些边，令  $w(e)$  表示边  $e$  的颜色（颜色从 1 到  $m$  编号）。对于所有整数  $1 \leq c \leq m$ ， $c$  在序列  $w(e_{u,1}), w(e_{u,2}), \dots, w(e_{u,d_u})$  中至多出现两次。

令  $f_u$  表示与  $u$  相邻的所有边中一共出现了几种不同的颜色。求一种给边涂色的方案，最小化  $\sum_{u=1}^n f_u$ 。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 10^4$ ) 表示测试数据组数，对于每组测试数据：

第一行输入两个整数  $n$  和  $m$  ( $2 \leq n \leq 2 \times 10^5$ ,  $1 \leq m \leq \min(\frac{n(n-1)}{2}, 2 \times 10^5)$ )，表示图中节点的数量和边的数量。

对于接下来  $m$  行，第  $i$  行输入两个整数  $u_i$  和  $v_i$  ( $1 \leq u_i, v_i \leq n$ )，表示第  $i$  条边连接节点  $u_i$  和  $v_i$ 。

保证图中没有自环或重边。然而，给定的图可能是不连通的。另外保证所有数据  $n$  之和与  $m$  之和均不超过  $2 \times 10^5$ 。

### Output

每组数据输出一行  $m$  个由单个空格分隔的整数  $w_1, w_2, \dots, w_m$  ( $1 \leq w_i \leq m$ )，其中  $w_i$  表示第  $i$  条边上涂的颜色。

如果有多种答案，您可以输出任意一种。

### Example

| standard input | standard output |
|----------------|-----------------|
| 2              | 2 2 3 3 2 3 6   |
| 5 7            | 2 1             |
| 1 5            |                 |
| 2 5            |                 |
| 3 5            |                 |
| 4 5            |                 |
| 1 2            |                 |
| 3 4            |                 |
| 3 1            |                 |
| 4 2            |                 |
| 1 2            |                 |
| 3 4            |                 |

## Problem F. 背包

有  $n$  组物品，第  $i$  组物品共有  $a_i$  个，每个物品的重量为  $2^{b_i}$ 。

有  $m$  个背包，每个背包的承重为  $k$ 。求最小的  $k$ ，使得所有  $\sum_{i=1}^n a_i$  个物品都能被放入背包中，且每个背包里物品的总重量都不超过  $k$ 。

请注意，每个物品都要被放入恰好一个背包。一个背包里可以有来自不同组的物品，来自同一组的物品也可以被放入不同背包中。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 10^4$ ) 表示测试数据组数，对于每组测试数据：

第一行输入两个整数  $n$  和  $m$  ( $1 \leq n \leq 2 \times 10^5$ ,  $1 \leq m \leq 10^9$ ) 表示物品的组数以及背包的数量。

对于接下来  $n$  行，第  $i$  行输入两个整数  $a_i$  和  $b_i$  ( $1 \leq a_i \leq 10^9$ ,  $0 \leq b_i \leq 10^9$ )，其中  $a_i$  是第  $i$  组物品的数量， $2^{b_i}$  是第  $i$  组里每个物品的重量。

保证所有数据  $n$  之和不超过  $2 \times 10^5$ 。

### Output

每组数据输出一行一个整数表示答案。由于答案可能很大，请将答案对 998 244 353 取模后输出。请注意，取模只是为了简化输出。您需要最小化的是取模之前的答案。

### Example

| standard input        | standard output |
|-----------------------|-----------------|
| 2                     | 10              |
| 5 4                   | 628956724       |
| 3 0                   |                 |
| 2 3                   |                 |
| 3 1                   |                 |
| 1 3                   |                 |
| 2 1                   |                 |
| 2 20250427            |                 |
| 1000000000 1000000000 |                 |
| 114514 1919810        |                 |

## Problem G. 路径求和问题

有一个  $n$  行  $m$  列的网格。网格里的每个格子都写着一个整数，其中第  $i$  行第  $j$  列的格子里写着整数  $a_{i,j}$ 。

令  $(i, j)$  表示位于第  $i$  行第  $j$  列的格子。您现在需要从  $(1, 1)$  出发并前往  $(n, m)$ 。当您位于格子  $(i, j)$  时，您可以选择走到右方的格子  $(i, j + 1)$ （若  $j < m$ ），也可以选择走到下方的格子  $(i + 1, j)$ （若  $i < n$ ）。

令  $S$  表示路径上每个格子里的整数形成的集合，包括  $a_{1,1}$  和  $a_{n,m}$ 。路径的价值定义为  $S$  中元素的数量（请回忆：集合中不包含重复元素）。对于所有可能的路径，求它们的价值之和。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 10^3$ ) 表示测试数据组数。对于每组测试数据：

第一行输入两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 10^5$ ,  $1 \leq n \times m \leq 10^5$ ) 表示网格的行数和列数。

对于接下来  $n$  行，第  $i$  行输入  $m$  个整数  $a_{i,1}, a_{i,2}, \dots, a_{i,m}$  ( $1 \leq a_{i,j} \leq n \times m$ )，其中  $a_{i,j}$  表示格子  $(i, j)$  里的整数。

保证所有数据  $n \times m$  之和不超过  $10^5$ 。

### Output

每组数据输出一行一个整数，表示所有可能的路径的价值之和。由于答案可能很大，请将答案对 998 244 353 取模后输出。

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | 7               |
| 2 3            | 1               |
| 5 2 1          | 3               |
| 1 5 5          |                 |
| 1 1            |                 |
| 1              |                 |
| 2 3            |                 |
| 3 3 3          |                 |
| 3 3 3          |                 |

### Note

对于第一组样例数据，有 3 条可能的路径：

- 第一条路径是  $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3)$ 。  $S = \{1, 2, 5\}$ 。
- 第二条路径是  $(1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow (2, 3)$ 。  $S = \{2, 5\}$ 。
- 第三条路径是  $(1, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (2, 3)$ 。  $S = \{1, 5\}$ 。

所以答案是  $3 + 2 + 2 = 7$ 。

## Problem H. 野火哥，你行不行？

Foammm 出了一道这样的题。

### 互质问题

给定两个正整数  $x$  和  $y$ 。可以执行以下四种操作任意次，每次花费 1 的代价：

- 将  $x$  加 1。
- 将  $y$  加 1。
- 将  $x$  减 1。
- 将  $y$  减 1。

您需要使这两个整数没有大于 1 的共同因子，求最小总代价。

Foammm 需要为该问题生成一些测试数据。她给了您一个整数  $k$ ，您需要找到两个正整数  $x$  和  $y$  使得上述问题的答案正好是  $k$ 。

### Input

每个测试文件只有一组测试数据。

第一行输入一个整数  $k$  ( $0 \leq k \leq 20$ )。

### Output

第一行输出一个整数  $x$  ( $0 < x < 10^{1500}$ )，第二行输出一个整数  $y$  ( $0 < y < 10^{1500}$ )。

可以证明答案总是存在。如果有多个合法答案，您可以输出其中任何一个。

### Examples

| standard input | standard output |
|----------------|-----------------|
| 0              | 1<br>1          |
| 1              | 2<br>2          |
| 2              | 945<br>1210     |

### Note



## Problem I. Bingo 3

本题中，您需要构造一个  $n$  行  $n$  列的网格。网格里的每个格子都写着一个整数，其中第  $i$  行第  $j$  列的格子里写着整数  $a_{i,j}$ 。从 1 到  $n^2$  的每个整数（含两端）在网格里都恰好出现一次。

称整数  $x$  是网格的“bingo 整数”，若以下两个条件至少满足一个。

- 至少存在一行，使得那一行所有格子里的整数都小于等于  $x$ 。
- 至少存在一列，使得那一列所有格子里的整数都小于等于  $x$ 。

容易发现，一个网格可以有很多 bingo 整数。不过本题中，我们只对最小的 bingo 整数感兴趣。

给定整数  $n$  和  $k$ ，构造一个  $n$  行  $n$  列的网格，使得其最小的 bingo 整数恰为  $k$ 。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 50$ ) 表示测试数据组数，对于每组测试数据：

第一行输入两个整数  $n$  和  $k$  ( $1 \leq n \leq 50$ ,  $1 \leq k \leq n^2$ )。

### Output

对于每组数据：

- 如果可以构造一个  $n$  行  $n$  列的网格，使得其最小的 bingo 整数恰为  $k$ ，首先输出一行 **Yes**。之后输出  $n$  行，第  $i$  行包含  $n$  个由单个空格分隔的整数  $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ，表示出现在网格第  $i$  行的整数。别忘了从 1 到  $n^2$  的每个整数（含两端）在网格里必须都恰好出现一次。如果有多种答案，您可以输出任意一种。
- 如果答案不存在，只要输出一行 **No**。

### Example

| standard input | standard output |
|----------------|-----------------|
| 4              | Yes             |
| 3 5            | 4 2 5           |
| 4 10           | 7 1 9           |
| 5 2            | 8 6 3           |
| 1 1            | Yes             |
|                | 14 9 2 13       |
|                | 1 11 16 8       |
|                | 10 3 7 5        |
|                | 6 15 4 12       |
|                | No              |
|                | Yes             |
|                | 1               |

## Problem J. 字典

您将在接下来的  $q$  天里从字典里学习单词。字典是一个长度为  $n$  的由小写字母构成的字符串  $S = s_1s_2\cdots s_n$ ，一个单词是  $S$  的一个子串。

在第  $i$  天，您将会学习所有以  $S[l_i:r_i] = s_{l_i}s_{l_i+1}\cdots s_{r_i}$  为开头的单词（也就是说， $S[l_i:r_i]$  是这些单词的前缀）。在每天结束后，求从第一天开始您一共已经学习了几个不同的单词。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 10^4$ ) 表示测试数据组数，对于每组测试数据：

第一行输入一个由小写字母构成的长度为  $n$  的字符串  $s_1s_2\cdots s_n$  ( $1 \leq n \leq 2 \times 10^5$ ) 表示字典。

第二行输入一个整数  $q$  ( $1 \leq q \leq 2 \times 10^5$ )，表示您将要学习单词的天数。

对于接下来  $q$  行，第  $i$  行输入两个整数  $l_i$  和  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ )，表示在第  $i$  天，您将会学习所有以  $s_{l_i}s_{l_i+1}\cdots s_{r_i}$  为前缀的单词。

保证所有数据  $n$  之和与  $q$  之和均不超过  $2 \times 10^5$ 。

### Output

每组数据输出一行  $q$  个由单个空格分隔的整数  $c_1, c_2, \dots, c_q$ ，其中  $c_i$  表示您从第 1 天到第  $i$  天（含两端）一共学习了几个不同的单词。

### Example

| standard input | standard output |
|----------------|-----------------|
| 2              | 4 6 7           |
| abcabd         | 3 3 3           |
| 3              |                 |
| 1 3            |                 |
| 1 2            |                 |
| 6 6            |                 |
| aaa            |                 |
| 3              |                 |
| 3 3            |                 |
| 2 3            |                 |
| 1 3            |                 |

### Note

对于第一组样例数据：

- 在第 1 天，您学习了 abc, abca, abcab, 以及 abcabd。所以答案是 4。
- 在第 2 天，您学习了 ab, abc, abca, abcab, abcabd, 以及 abd。其中有 4 个单词已经学过了，所以答案是  $4 + (6 - 4) = 6$ 。
- 在第 3 天，您学习了 d，这个单词之前没学过。所以答案是  $6 + 1 = 7$ 。

对于第二组样例数据，您在第 1 天就已经学习了所有单词 (a, aa, aaa)，所以答案总是 3。

## Problem K. 拉斯维加斯

《拉斯维加斯》是一款简单易学的骰子类桌游。玩家投掷骰子，并将骰子放置到某座赌场中。每座赌场中拥有最多骰子，且与其他玩家平局的人赢得该赌场的金钱。这款游戏在运气与策略中做出了完美的平衡，并获得了 2012 德国年度游戏奖的提名。



BoardGameGeek 用户 @denislageron 拍摄的照片

考虑一局有  $n$  座赌场以及  $(m + 1)$  名玩家的游戏。您是第  $(m + 1)$  位玩家，其他玩家都已经在赌场中放好了（或决定不放）骰子。已知在第  $i$  座赌场中，第  $j$  名玩家放置了  $a_{i,j}$  颗骰子。

对于每个  $1 \leq i \leq n$ ，您需要决定在第  $i$  座赌场中要放置的骰子数量  $b_i$ 。在您放置完骰子后，我们将根据以下规则分别决出每座赌场的赢家：

1. 首先，若两名或以上玩家在该赌场中放置了相同数量的骰子，他们在该赌场里的骰子会被全部移除。
2. 之后，在该赌场里拥有最多骰子的玩家成为该赌场的赢家。若赌场里没有骰子，则没有赢家。

例如：

- 考虑 7 名玩家在一座赌场中分别放置了 1, 2, 3, 3, 4, 4, 4 颗骰子。首先，因为有两名或以上玩家都放置了 3 颗或 4 颗骰子，他们的骰子都会被移除，剩下的骰子数分别为 1, 2, 0, 0, 0, 0, 0。该赌场的赢家是放置了 2 颗骰子的玩家。
- 考虑 4 名玩家在一座赌场中分别放置了 2, 3, 2, 3 颗骰子。首先，因为有两名或以上玩家都放置了 2 颗或 3 颗骰子，他们的骰子都会被移除，剩下的骰子数分别为 0, 0, 0, 0。由于该赌场中没有骰子，因此没有赢家。

您的目标是和所有玩家相比，在最多的赌场中成为赢家。更正式地，设  $w_i$  表示玩家  $i$  在几座赌场里成为了赢家，则对于所有  $1 \leq i \leq m$ ，都必须有  $w_{m+1} \geq w_i$ 。您需要在达成目标的同时，最小化放置骰子的总数。也就是说，最小化  $\sum_{i=1}^n b_i$ 。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 100$ ) 表示测试数据组数，对于每组测试数据：

第一行输入两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 50$ )，表示赌场的数量和（除了您以外的）玩家数量。

对于接下来  $n$  行，第  $i$  行输入  $m$  个整数  $a_{i,1}, a_{i,2}, \dots, a_{i,m}$  ( $0 \leq a_{i,j} \leq 10^8$ )，其中  $a_{i,j}$  表示赌场  $i$  中玩家  $j$  放置的骰子数量。

保证至多有 3 组数据满足  $n > 20$  或  $m > 20$ 。

## Output

每组数据输出一行  $n$  个由单个空格分隔的整数  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i \leq 10^9$ )，其中  $b_i$  表示您在赌场  $i$  中放置的骰子数量。可以证明在对  $b_i$  的此种限制下，总是存在一个最佳答案。如果有多种最佳答案，您可以输出任意一种。

## Example

| standard input | standard output |
|----------------|-----------------|
| 3              | 4 0 6 0         |
| 4 3            | 1 0 2           |
| 3 3 2          | 0               |
| 2 7 5          |                 |
| 3 5 2          |                 |
| 1 6 4          |                 |
| 3 4            |                 |
| 100 100 100 1  |                 |
| 0 0 0 1        |                 |
| 100 100 100 1  |                 |
| 1 4            |                 |
| 20 0 20 0      |                 |

## Note

对于第一组样例数据，您是赌场 1 和 3 的赢家，而玩家 2 是赌场 2 和 4 的赢家。

对于第二组样例数据，您是赌场 3 的赢家，而玩家 4 是赌场 2 的赢家。赌场 1 没有赢家。

## Problem L. 子序列

称一个长度为  $k$  的整数序列  $C = c_1, c_2, \dots, c_k$  是好的，若序列里最小值与最大值的平均值等于序列的中位数。长度为  $k$  的序列的中位数定义为序列里第  $\lceil \frac{k}{2} \rceil$  小的元素，其中  $\lceil x \rceil$  表示大于等于  $x$  的最小整数。例如，序列 1, 7, 4, 3 的中位数是 3，而序列 5, 8, 2, 1, 6 的中位数是 5。

更正式地，令  $D = d_1, d_2, \dots, d_k$  表示将序列  $C$  从小到大排序后得到的序列。称  $C$  是好的，若

$$\frac{d_1 + d_k}{2} = d_{\lceil \frac{k}{2} \rceil}$$

给定整数序列  $A = a_1, a_2, \dots, a_n$ ，求其最长好子序列的长度。请回忆，称序列  $B$  是序列  $A$  的子序列，若  $B$  可以通过以下方式获得：从  $A$  中删除零个或多个元素，同时不改变剩余元素的相对顺序。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 300$ ) 表示测试数据组数，对于每组测试数据：

第一行输入一个整数  $n$  ( $1 \leq n \leq 3 \times 10^3$ ) 表示序列的长度。

第二行输入  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) 表示序列。

保证所有数据  $n$  之和不超过  $3 \times 10^3$ 。

### Output

每组数据输出一行一个整数，表示最长好子序列的长度。

### Example

| standard input   | standard output |
|------------------|-----------------|
| 4                | 5               |
| 7                | 4               |
| 3 5 9 8 2 11 5   | 1               |
| 7                | 2               |
| 7 9 2 4 17 10 15 |                 |
| 1                |                 |
| 100              |                 |
| 2                |                 |
| 100 100          |                 |

### Note

对于第一组样例数据，最长好子序列是 3, 5, 8, 2, 5。它的最小元素是 2，最大元素是 8，中位数是 5。

对于第二组样例数据，最长好子序列是 7, 9, 4, 10。它的最小元素是 4，最大元素是 10，中位数是 7。

## Problem M. 航班追踪器

假设地球是三维欧几里得空间里球心为  $(0,0,0)$  且半径为  $r$  的球。一架飞机正沿着地球表面的最短路径从出发地飞往目的地。

作为一名航空爱好者，您有一个接收器，可以接收到距离不超过  $d$  的飞机的信号。请注意，两点间的距离是按地球表面最短路径的长度计算的，而不是三维欧几里得空间中的欧几里得距离。求最小的  $d$ ，使得在某个时刻，您可以从您的房子里接收到这架飞机的信号。

### Input

有多组测试数据。第一行输入一个整数  $T$  ( $1 \leq T \leq 10^4$ ) 表示测试数据组数，对于每组测试数据：

第一行输入一个整数  $r$  ( $1 \leq r \leq 100$ ) 表示地球的半径。

第二行输入三个整数  $a, b$  和  $c$  ( $-100 \leq a, b, c \leq 100, a^2 + b^2 + c^2 > 0$ )，表示您的房子的坐标是  $\left(\frac{ra}{\sqrt{a^2+b^2+c^2}}, \frac{rb}{\sqrt{a^2+b^2+c^2}}, \frac{rc}{\sqrt{a^2+b^2+c^2}}\right)$ 。

第三行输入三个整数  $u, v$  和  $w$  ( $-100 \leq u, v, w \leq 100, u^2 + v^2 + w^2 > 0$ )，表示出发地的坐标是  $\left(\frac{ru}{\sqrt{u^2+v^2+w^2}}, \frac{rv}{\sqrt{u^2+v^2+w^2}}, \frac{rw}{\sqrt{u^2+v^2+w^2}}\right)$ 。

第四行输入三个整数  $x, y$  和  $z$  ( $-100 \leq x, y, z \leq 100, x^2 + y^2 + z^2 > 0$ )，表示目的地的坐标是  $\left(\frac{rx}{\sqrt{x^2+y^2+z^2}}, \frac{ry}{\sqrt{x^2+y^2+z^2}}, \frac{rz}{\sqrt{x^2+y^2+z^2}}\right)$ 。

保证出发地与目的地既不会重合，也不会位于地球的两端。因此，出发地与目的地之间的最短路径是唯一确定的。

### Output

每组数据输出一行一个实数，表示最小的  $d$ ，使得在某个时刻，您可以从您的房子里接收到这架飞机的信号。

如果您的答案的绝对误差或相对误差不超过  $10^{-4}$ ，则将被视为正确。更正式地，假设您的输出为  $a$ ，标准答案为  $b$ ，当且仅当  $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-4}$  时，您的输出才会被接受。

## Example

| standard input | standard output        |
|----------------|------------------------|
| 2              | 61.547970867038734110  |
| 100            | 235.619449019234492887 |
| 1 1 1          |                        |
| 1 0 0          |                        |
| 0 1 0          |                        |
| 100            |                        |
| -1 -1 0        |                        |
| 1 0 0          |                        |
| 0 1 0          |                        |

## Note

下图展示了第一组样例数据。 $P$  是您的房子， $S$  是出发地， $T$  是目的地。

