

2026 克露丝卡尔杯程序设计竞赛



2026 年 9 月 13 日

本试题册共 13 题，19 页。

出题组 & 主办方

中国科学技术大学校学生计算机编程俱乐部





Problem A. 掐头去尾

给定一个长度为 n 的数组 a 。对于每个 $k = 1, 2, \dots, n$ ，独立地考虑以下过程。
初始时，数组为 a 。你需要恰好进行 k 次操作。每次操作可以选择以下两种方式之一：

- **Backspace**: 删除当前数组的第一个元素；
- **Delete**: 删除当前数组的最后一个元素。

你的得分定义为第 k 次操作中被删除元素的值。
对于每个 $k = 1, 2, \dots, n$ ，求你能够获得的最大得分。

Input

第一行包含一个整数 n ，表示数组的长度 ($1 \leq n \leq 10^5$)。
第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ，表示数组 a ($1 \leq a_i \leq 10^9$)。

Output

输出 n 个整数。其中，第 k 个整数表示恰好进行 k 次操作时能够获得的最大得分。

Example

standard input	standard output
5 2 7 8 1 4	4 7 8 8 8

Note

对于 $k = 1$ ，只能删除数组最左边的 2 或最右边的 4，因此最大得分为 4。
对于 $k = 2$ ，可以第一次删除最左边的 2，第二次再删除最左边的 7，此时第二次操作删除的元素为 7，因此最大得分为 7。
对于 $k = 3$ ，可以依次删除最左边的 2, 7, 8，使第三次操作删除的元素为 8，因此最大得分为 8。
对于 $k = 4$ 和 $k = 5$ ，同样可以合理安排前面的操作，使最后一次操作删除的元素为 8。



Problem B. 消失的逆序对

Alice 和 Bob 在一个排列上进行游戏, Alice 先手。

称一个长度为 n 的序列为一个**排列**, 当且仅当 $1, 2, \dots, n$ 中的每个整数在序列中恰好出现一次。

初始给定一个长度为 n 的排列 $a_1, a_2, \dots, a_n$ 。在游戏过程中的任意时刻, 当前序列始终为一个长度为 m 的排列。双方轮流操作, 每次必须选择下列两种操作之一:

- 选择一个满足 $a_i > a_{i+1}$ 的下标 i ($1 \leq i < m$), 并交换 a_i 和 a_{i+1} ;
- 选择一个满足 $a_{i+1} = a_i + 1$ 的下标 i ($1 \leq i < m$), 并删除 a_i 和 a_{i+1} , 随后将剩余元素按照相对大小重新编号: 其中第 k 小的元素被重新编号为 k , 使得剩余序列重新成为一个排列。

例如, 当前排列为 $[2, 3, 4, 1]$ 。可以删除相邻的 $2, 3$, 剩余序列为 $[4, 1]$; 重新编号后, 排列变为 $[2, 1]$ 。

当一名玩家无法进行任何操作时, 该玩家输掉游戏。假设 Alice 和 Bob 都采用最优策略, 请判断最终获胜者。

Input

本题有多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 5000$)。

对于每组测试数据:

- 第一行包含一个整数 n ($1 \leq n \leq 5000$);
- 第二行包含 n 个整数 $a_1, a_2, \dots, a_n$, 保证 a 是一个长度为 n 的排列。

保证所有测试数据的 n 之和不超过 5000。

Output

对于每组测试数据, 如果 Alice 获胜, 输出 Alice; 否则输出 Bob。

Example

standard input	standard output
5	Bob
1	Alice
1	Bob
2	Alice
1 2	Bob
2	
2 1	
4	
2 4 1 3	
5	
5 1 4 2 3	



Problem C. 极差数组操作

一个数组的**极差**定义为其最大元素与最小元素之差。

给定一个长度为 n 的整数数组 a 。你可以执行若干次以下操作（也可以不执行）：

- 选择两个不同的下标 i, j ($1 \leq i, j \leq n$ 且 $i \neq j$)，然后按以下操作赋值：

$$a_i \leftarrow a_i + 3, \quad a_j \leftarrow a_j - 1.$$

请计算经过若干次操作后，数组 a 的极差的**最小值**。

Input

本题有多组测试数据。

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。

对于每组测试用例：

- 第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示数组 a 的长度；
- 第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$)，表示数组 a 中的元素。

保证所有测试用例的 n 之和不超过 2×10^5 。

Output

对于每组测试用例，输出一行一个整数，表示经过任意有限次操作后能够得到的最小极差。

Example

standard input	standard output
3	0
1	2
5	1
2	
0 2	
4	
-1 9 5 8	

Note

第一组测试用例只有一个元素，因此极差恒为 0。



Problem D. 长廊调律

一条长廊上依次放置了 n 个共振器。第 i 个共振器当前的频率偏移量为 a_i ，目标偏移量为 b_i 。调律装置只能从长廊的一端启动。一次操作中，你可以任选一个非零整数 x ，并执行以下两种操作之一：

1. 选择一个 k ($1 \leq k \leq n$)，将前缀 $a_1, a_2, \dots, a_k$ 中的每个数都加上 x ；
2. 选择一个 k ($1 \leq k \leq n$)，将后缀 $a_k, a_{k+1}, \dots, a_n$ 中的每个数都加上 x 。

x 可以为正数或负数。无论 $|x|$ 多大，本次修改都只计为一次操作。

求将序列 a 变为序列 b 所需的最少操作次数。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例数量。

对于每个测试用例：

- 第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)；
- 第二行 n 个整数 $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$)；
- 第三行 n 个整数 $b_1, b_2, \dots, b_n$ ($-10^9 \leq b_i \leq 10^9$)。

保证所有测试用例的 n 之和不超过 2×10^5 ，所有测试用例的 $\sum |a_i - a_{i-1}| (i \geq 2)$ 之和与 $\sum |b_i - b_{i-1}| (i \geq 2)$ 之和分别不超过 10^4 。

Output

对于每个测试用例，输出一行一个整数，表示最少操作次数。

Example

standard input	standard output
4	2
3	3
0 0 0	1
-1 2 0	0
3	
0 0 0	
3 1 4	
4	
1 4 2 8	
6 9 7 13	
2	
-7 10	
-7 10	

Note

对于第一个测试用例，可以执行：

1. 给长度为 1 的前缀加上 -3 ，得到 $[-3, 0, 0]$ ；
2. 给长度为 2 的前缀加上 2，得到 $[-1, 2, 0]$ 。

对于第三个测试用例，给整个序列加上 5 即可。



Problem E. 去心邻域

这是一道交互题。

有一个未知序列 B ，元素是 0 或 1，下标从 1 开始。你可以进行多次询问，每次询问指定中心 m 以及半径 r ，交互器将返回 B 的下标去心邻域 $[m - r, m + r] \setminus \{m\}$ 内 1 的个数。尝试确定 B 中一共有多少个 1。

注意，你询问的区间不能超出序列本身的边界。

Interaction Protocol

本题有多组测试数据。

首先读入一行，仅包含一个正整数，表示数据组数 T ($1 \leq T \leq 1000$)。

对于每组数据：

- 首先读入一行，仅包含一个整数 n ($4 \leq n \leq 1000$)，表示 B 的长度。
- 发起询问时，需要以 `? m r` 的格式输出一行并**清空缓冲区**，要求 $2 \leq m \leq n - 1, r \geq 1, m - r \geq 1, m + r \leq n$ 。然后读入一行，包含一个非负整数，表示 $\sum_{i=1}^r (B_{m-i} + B_{m+i})$ 的值。
- 当你确定答案后，以 `! x` 的格式输出一行并**清空缓冲区**，其中 x 表示 B 中 1 的个数。
- 特别地，若无论如何询问都不可能确定答案，输出一行 `! -1` 并**清空缓冲区**。

每组数据的询问次数不能超过 35。

任何不符合交互要求的输出，包括交互格式错误、超出交互次数限制等，都会导致未定义的运行结果。

Example

standard input	standard output
2	
4	
	? 2 1
1	
	? 3 1
1	
	! 2
5	
	? 3 2
4	
	? 2 1
2	
	! 5

Note

样例展示了一个可能的交互过程，两组数据中的未知序列分别为 1100 和 11111。

如何**清空缓冲区**：

- 在 C 和 C++ 中，使用 `fflush(stdout)`（如果使用 `printf`）或 `cout.flush()`（如果使用 `cout`）；
- 在 Python 中，使用 `stdout.flush()`；
- 特别地，在 C++ 中，使用 `cout << endl` 会自动清空缓冲区。



Problem F. 简单构造题

长为 n 的“红专序列”指满足如下特征的序列：

- $1 \sim n$ 在序列中各出现恰好一次；
- 相邻两项的和均不是完全平方数。

给定整数 n ，构造一个长度为 n 的“红专序列”。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示数据组数。
对于每组数据，仅一行一个整数 n ($1 \leq n \leq 4 \times 10^6$)，表示需要构造的“红专序列”的长度。
保证 n 的总和不超过 4×10^6 ，可以证明在所给条件下一定存在一个解。

Output

输出 T 行，每行为需要构造的序列。如果有多个，输出任意一个即可。

Example

standard input	standard output
5	1
1	1 2
2	1 2 3
3	1 2 3 4
4	1 4 2 3 5
5	

Note

对于 $n = 5$ ，如果输出为 1 2 3 4 5 则是错误的，因为 $4 + 5 = 9$ 为完全平方数。



Problem G. 奶龙大战暴暴龙 3.1

Input file: standard input
Output file: standard output

这是一道通信题。在本题中，你的程序将运行两次。两次运行之间，内存中存储的所有变量都将丢失，但第一次运行中获取的信息可能对第二次运行中正确解决问题非常重要。

本题中有“奶龙”和“暴暴龙”两个角色。奶龙拥有一棵包含 n 个节点的树¹的完整结构信息，而暴暴龙只知道树的节点数 n 。由于暴暴龙被邪恶的小豹子关起来了，所以奶龙只能通过一种特殊的单向通信方式，来帮助暴暴龙还原出一棵与原树同构²的树。

通信的规则如下：

- 奶龙需要给树上的每个节点染色为黑色，白色或者灰色；
- 小豹子会按如下代码生成一个序列；
- 奶龙只能把这个长为 $2n - 1$ 的序列 $c_1, c_2, \dots, c_{2n-1}$ 发送给暴暴龙。

Algorithm 1 Encode($T, color$)

```
1: procedure DFS( $u, parent$ )
2:   append  $u$  to  $tour$ 
3:   for  $v$  in vertices adjacent to  $u$  do                                 $\triangleright v$  can be iterated in any order.
4:     if  $v \neq parent$  then
5:       DFS( $v, u$ )
6:       append  $u$  to  $tour$ 
7:     end if
8:   end for
9: end procedure
10: procedure GENERATESEQUENCE( $T, color$ )
11:    $tour \leftarrow$  empty sequence
12:    $root \leftarrow$  the starting vertex chosen by the leopard
13:   DFS( $root, NIL$ )
14:    $c \leftarrow$  empty sequence
15:   for  $v$  in  $tour$  do
16:     append  $color_v$  to  $c$ 
17:   end for
18:   return  $c$ 
19: end procedure
```

奶龙无法直接将节点编号发给暴暴龙，她只能将序列 c 发送给暴暴龙。暴暴龙在收到 n 以及序列 c 后，需要构造并输出一棵与奶龙的树同构的树。

Communication Protocol

每个测试点中，选手程序将被运行两次。在下发文件中，提供了一份测试工具供选手本地调试使用。

First Run

在第一次运行中，你将扮演“奶龙”角色。

¹一棵有 n 个节点的树可以表示为 $T(V, E)$ ，其中 $|V| = n, |E| = n - 1$ ，且 $E \subseteq \{\{u, v\} | u, v \in V\}$ ，且对于所有的 $u, v \in V, u \neq v$ ，存在 $u = p_0, p_1, \dots, p_k = v$ 且 $\{p_{i-1}, p_i\} \in E$ 对所有的 $i = 1, 2, \dots, k$ 成立。其中， V 中的元素称为顶点， E 中的元素称为边。节点 u 和节点 v 相邻当且仅当 $\{u, v\} \in E$ 。

²同构：称两棵树 $T_1(V_1, E_1)$ 和 $T_2(V_2, E_2)$ 同构，当且仅当存在双射 $f: V_1 \rightarrow V_2, \forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2$ 。



Input

输入的第一行包含一个字符串 `first`，其作用是让你的程序能够识别这是第一次运行。
第二行包含一个整数 T ($1 \leq T \leq 10^4$)，表示数据组数。
对于每组数据：

- 第一行包含一个整数 n ($2 \leq n \leq 2 \times 10^5$)，表示树的节点数。
- 接下来 $n - 1$ 行，每行包含两个整数 u, v ($1 \leq u, v \leq n$)，表示树上的一条边。

数据保证 n 的和不超过 2×10^5 。

Output

你的输出应该包含 T 行，第 i 行包含一个长度与第 i 棵树节点数量 n 相同的序列 $col_1, col_2, \dots, col_n$ ($col_i \in \{0, 1, 2\}, i = 1, 2, \dots, n$)，表示奶龙给第 i 个节点染色为 col_i 。 $col_i = 0$ 为白色， $col_i = 1$ 为黑色， $col_i = 2$ 为灰色。

Second Run

在第二次运行中，你将扮演“暴暴龙”角色。

Input

输入的第一行包含一个字符串 `second`，其作用是让你的程序能够识别这是第二次运行。
第二行包含一个整数 T ($1 \leq T \leq 10^4$)，表示数据组数。
对于每组数据：

- 第一行包含一个整数 n ($2 \leq n \leq 2 \times 10^5$)，表示树的节点数。
- 接下来一行包含 $2n - 1$ 个整数，表示评测机根据奶龙的染色生成的序列。

数据保证 n 的和不超过 2×10^5 。

Output

对于第 i 组数据，设输入节点数为 n ，则输出 $n - 1$ 行，每行包含两个整数 u, v ，表示你还原出的树的一条边 ($1 \leq u, v \leq n$)。你需要保证输出的树和第一次运行中输入的树同构。

Examples

standard input	standard output
first	0 1
2	0 1 2
2	
1 2	
3	
1 2	
2 3	
second	1 3
2	2 3
3	1 2
0 1 2 1 0	
2	
0 1 0	



Note

两个样例演示了同一测试点中的两次运行。

注意, 如果第一次输入的树依次为 $t_1, \dots, t_T$, 输出的颜色序列为 $col_1, \dots, col_T$ 。设评测器根据 t_i, col_i ($1 \leq i \leq T$) 得到的颜色序列为 a_i , 评测器会随机生成一个 $1 \sim T$ 的排列 $p_1, \dots, p_T$, 然后评测器发送到第二次输入的顺序为 $a_{p_1}, \dots, a_{p_T}$, 输出的树依次为 $t'_1, t'_2, \dots, t'_T$, 评测器会将 t_{p_i} 与 t'_i 比较进行评测。

下发文件使用方法

```
python3 tree_communication_testing_tool1.py data.in ./solution
python3 tree_communication_testing_tool1.py --trials 10 data.in ./solution
python3 tree_communication_testing_tool1.py data.in python3 solution.py
```

其中 `solution` 或 `solution.py` 为 C/C++ 编译出的可执行文件或 python 源码, `data.in` 为第一次输入的数据, `trials` 参数表示该重复测试的次数。



Problem H. 奶龙大战暴暴龙 3.2

Input file: standard input
Output file: standard output

这是一道通信题。在本题中，你的程序将运行两次。两次运行之间，内存中存储的所有变量都将丢失，但第一次运行中获取的信息可能对第二次运行中正确解决问题非常重要。

本题中有“奶龙”和“暴暴龙”两个角色。奶龙拥有一棵包含 n 个节点的树¹的完整结构信息，而暴暴龙只知道树的节点数 n 。由于暴暴龙被邪恶的小豹子关起来了，所以奶龙只能通过一种特殊的单向通信方式，来帮助暴暴龙还原出一棵与原树同构²的树。

通信的规则如下：

- 奶龙需要给树上的每个节点染色为黑色或者白色；
- 小豹子会按如下代码生成一个序列；
- 奶龙只能把这个长为 $2n - 1$ 的序列 $c_1, c_2, \dots, c_{2n-1}$ 发送给暴暴龙。

Algorithm 2 Encode($T, color$)

```
1: procedure DFS( $u, parent$ )
2:   append  $u$  to  $tour$ 
3:   for  $v$  in vertices adjacent to  $u$  do                                 $\triangleright v$  can be iterated in any order.
4:     if  $v \neq parent$  then
5:       DFS( $v, u$ )
6:       append  $u$  to  $tour$ 
7:     end if
8:   end for
9: end procedure
10: procedure GENERATESEQUENCE( $T, color$ )
11:    $tour \leftarrow$  empty sequence
12:    $root \leftarrow$  the starting vertex chosen by the leopard
13:   DFS( $root, NIL$ )
14:    $c \leftarrow$  empty sequence
15:   for  $v$  in  $tour$  do
16:     append  $color_v$  to  $c$ 
17:   end for
18:   return  $c$ 
19: end procedure
```

奶龙无法直接将节点编号发给暴暴龙，她只能将序列 c 发送给暴暴龙。暴暴龙在收到 n 以及序列 c 后，需要构造并输出一棵与奶龙的树同构的树。

Communication Protocol

每个测试点中，选手程序将被运行两次。在下发文件中，提供了一份测试工具供选手本地调试使用。

First Run

在第一次运行中，你将扮演“奶龙”角色。

¹一棵有 n 个节点的树可以表示为 $T(V, E)$ ，其中 $|V| = n, |E| = n - 1$ ，且 $E \subseteq \{\{u, v\} | u, v \in V\}$ ，且对于所有的 $u, v \in V, u \neq v$ ，存在 $u = p_0, p_1, \dots, p_k = v$ 且 $\{p_{i-1}, p_i\} \in E$ 对所有的 $i = 1, 2, \dots, k$ 成立。其中， V 中的元素称为顶点， E 中的元素称为边。节点 u 和节点 v 相邻当且仅当 $\{u, v\} \in E$ 。

²同构：称两棵树 $T_1(V_1, E_1)$ 和 $T_2(V_2, E_2)$ 同构，当且仅当存在双射 $f: V_1 \rightarrow V_2, \forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2$ 。



Input

输入的第一行包含一个字符串 `first`，其作用是让你的程序能够识别这是第一次运行。
第二行包含一个整数 T ($1 \leq T \leq 10^4$)，表示数据组数。
对于每组数据：

- 第一行包含一个整数 n ($2 \leq n \leq 2 \times 10^5$)，表示树的节点数。
- 接下来 $n - 1$ 行，每行包含两个整数 u, v ($1 \leq u, v \leq n$)，表示树上的一条边。

数据保证 n 的和不超过 2×10^5 。

Output

你的输出应该包含 T 行，第 i 行包含一个长度与第 i 棵树节点数量 n 相同的序列 $col_1, col_2, \dots, col_n$ ($col_i \in \{0, 1\}, i = 1, 2, \dots, n$)，表示奶龙给第 i 个节点染色为 col_i 。 $col_i = 0$ 为白色， $col_i = 1$ 为黑色。

Second Run

在第二次运行中，你将扮演“暴暴龙”角色。

Input

输入的第一行包含一个字符串 `second`，其作用是让你的程序能够识别这是第二次运行。
第二行包含一个整数 T ($1 \leq T \leq 10^4$)，表示数据组数。
对于每组数据：

- 第一行包含一个整数 n ($2 \leq n \leq 2 \times 10^5$)，表示树的节点数。
- 接下来一行包含 $2n - 1$ 个整数，表示评测机根据奶龙的染色生成的序列。

数据保证 n 的和不超过 2×10^5 。

Output

对于第 i 组数据，设输入节点数为 n ，则输出 $n - 1$ 行，每行包含两个整数 u, v ，表示你还原出的树的一条边 ($1 \leq u, v \leq n$)。你需要保证输出的树和第一次运行中输入的树同构。

Examples

standard input	standard output
first	0 1
2	0 1 0
2	
1 2	
3	
1 2	
2 3	
second	1 3
2	2 3
3	1 2
0 1 0 1 0	
2	
0 1 0	



Note

两个样例演示了同一测试点中的两次运行。

注意，如果第一次输入的树依次为 $t_1, \dots, t_T$ ，输出的颜色序列为 $col_1, \dots, col_T$ 。设评测器根据 t_i, col_i ($1 \leq i \leq T$) 得到的颜色序列为 a_i ，评测器会随机生成一个 $1 \sim T$ 的排列 $p_1, \dots, p_T$ ，然后评测器发送到第二次输入的顺序为 $a_{p_1}, \dots, a_{p_T}$ ，输出的树依次为 $t'_1, t'_2, \dots, t'_T$ ，评测器会将 t_{p_i} 与 t'_i 比较进行评测。

下发文件使用方法

```
python3 tree_communication_testing_tool2.py data.in ./solution
python3 tree_communication_testing_tool2.py --trials 10 data.in ./solution
python3 tree_communication_testing_tool2.py data.in python3 solution.py
```

其中 `solution` 或 `solution.py` 为 C/C++ 编译出的可执行文件或 python 源码，`data.in` 为第一次输入的数据，`trials` 参数表示该重复测试的次数。



Problem I. 同频回声

给定一棵以节点 1 为根、包含 n 个节点的带权树¹。

节点 i 具有：

- 频段 c_i ;
- 发射时刻 a_i 。

每个频段 c 具有重要度 w_c 。

对于两个使用相同频段的不同节点 u, v ，定义它们的同步代价为

$$D(u, v) = |a_u - a_v| + \text{dist}(u, v),$$

其中 $\text{dist}(u, v)$ 表示 u, v 之间简单路径²上的边权之和。

一次询问给出节点 x 和非负整数 K 。

称频段 c 在 x 的管辖区域中产生了回声，当且仅当存在两个不同节点

$$u, v \in \text{subtree}(x)$$

满足

$$c_u = c_v = c, \quad D(u, v) \leq K。$$

其中 $\text{subtree}(x)$ 表示以 x 为根的子树³。

对于每次询问，求所有产生回声的频段的重要度之和，即

$$\sum_{c=1}^m w_c [\exists u \neq v \in \text{subtree}(x), c_u = c_v = c, D(u, v) \leq K]。$$

方括号表示 Iverson bracket：条件成立时取 1，否则取 0。

Input

第一行包含三个整数 n, m, q ($1 \leq n \leq 10^6, 1 \leq m \leq 10^5, 1 \leq q \leq 10^6$)，分别表示节点数、频段数和询问数。

第二行包含 m 个整数 $w_1, w_2, \dots, w_m$ ($0 \leq w_c \leq 10^9$)，表示各个频段的重要度。

第三行包含 n 个整数 $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq m$)，表示各节点使用的频段。

第四行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$)，表示各节点的发射时刻。

接下来 $n - 1$ 行，每行包含三个整数 u, v, d ($1 \leq u, v \leq n, 0 \leq d \leq 10^9$)，表示节点 u, v 之间存在一条边权为 d 的无向边。

接下来 q 行，每行包含两个整数 x, K ($1 \leq x \leq n, 0 \leq K \leq 4 \times 10^{18}$)，表示一次询问。

保证输入的边构成一棵树，且答案能够用有符号 64 位整数表示。

Output

对于每次询问，输出一行一个整数，表示答案。

¹一棵有 n 个节点的树可以表示为 $T(V, E)$ ，其中 $|V| = n, |E| = n - 1$ ，且 $E \subseteq \{\{u, v\} | u, v \in V\}$ ，且对于所有的 $u, v \in V, u \neq v$ ，存在 $u = p_0, p_1, \dots, p_k = v$ 且 $\{p_{i-1}, p_i\} \in E$ 对所有的 $i = 1, 2, \dots, k$ 成立。其中， V 中的元素称为顶点， E 中的元素称为边。节点 u 和节点 v 相邻当且仅当 $\{u, v\} \in E$ 。

²这里 u, v 之间的简单路径指的是 $u = p_0, \dots, p_k = v$ ，且 $\forall i \neq j, p_i \neq p_j$ ，且 $\{p_{i-1}, p_i\} \in E$ 对 $\forall i = 1, 2, \dots, k$ 成立。

³ v 在 u 的子树中当且仅当 u 在根到 v 的简单路径上。



Example

standard input	standard output
7 3 5	0
5 7 11	12
1 2 1 3 2 1 3	7
10 4 13 8 9 20 7	5
1 2 2	23
1 3 3	
2 4 4	
2 5 1	
3 6 2	
3 7 5	
1 5	
1 6	
2 6	
3 9	
1 15	

Note

同频节点之间的同步代价如下：

- 频段 1 的节点为 1,3,6: $D(1,3) = |10 - 13| + 3 = 6$, $D(1,6) = |10 - 20| + 5 = 15$, $D(3,6) = |13 - 20| + 2 = 9$;
- 频段 2 的节点为 2,5: $D(2,5) = |4 - 9| + 1 = 6$;
- 频段 3 的节点为 4,7: $D(4,7) = |8 - 7| + 14 = 15$ 。

因此：

- 询问 (1,5) 中没有频段产生回声， 答案为 0;
- 询问 (1,6) 中频段 1,2 产生回声， 答案为 $5 + 7 = 12$;
- 询问 (2,6) 中只有频段 2 产生回声， 答案为 7;
- 询问 (3,9) 中只有频段 1 产生回声， 答案为 5;
- 询问 (1,15) 中三个频段均产生回声， 答案为 $5 + 7 + 11 = 23$ 。



Problem J. 文回文

给定长度为 n 的字符串 s ，下标从 1 开始。对于每个 $1 \leq i \leq n$ ，如果以 i 为中心，长度为 $2r + 1$ 的子串是回文的，则称非负整数 r 是中心 i 的一个回文半径。
对每个中心 i ，找到使对应回文子串的字典序最小的回文半径。

Input

本题有多组测试数据。
第一行包含一个整数 T ($1 \leq T \leq 10^6$)，表示测试数据组数。
每组数据一行，包含一个非空字符串 s 。保证 s 仅由小写英文字母构成。
保证 $\sum |s| \leq 10^6$ 。

Output

输出 T 行，每行输出 $|s|$ 个整数，其中第 i 个整数表示中心 i 的答案。

Example

standard input	standard output
2	0 0 1 0 0 0 0
cabacbc	0
a	

Note

样例第一组测试中，以 3 为中心的回文子串共有三个，分别是 b、aba、cabac，其中字典序最小者为 aba，半径为 1。



Problem K. 标准卡包

给定两个互质的正整数 p 和 q 。现有 pq 种不同的卡片，编号分别为 $0, 1, \dots, pq - 1$ 。

游戏中有两类标准卡包：

- 第一类标准卡包：共有 p 种，编号为 u ($0 \leq u < p$)。编号为 u 的卡包内包含所有编号满足 $x \equiv u \pmod{p}$ 的卡片各一张（每包共 q 张）。
- 第二类标准卡包：共有 q 种，编号为 v ($0 \leq v < q$)。编号为 v 的卡包内包含所有编号满足 $x \equiv v \pmod{q}$ 的卡片各一张（每包共 p 张）。

现在桌面上排列着 n 张卡片，第 i 张卡片的编号为 a_i 。你需要处理 Q 次操作，操作分为以下两种：

- **1 i x**：将第 i 张卡片的编号替换为 x ；
- **2 l r**：判断区间 $[l, r]$ 内的这些卡片，能否恰好被划分成若干个标准卡包。

注：在重新打包时，区间内的每一张卡片都必须恰好被分入一个卡包中，每一种标准卡包都可以被使用任意多次。

Input

第一行包含四个整数 n, p, q, Q ($1 \leq n, p, q, Q \leq 3 \times 10^5$, $2 \leq p + q \leq 3 \times 10^5$, $\gcd(p, q) = 1$)。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i < pq$)，表示初始时每张卡片的编号。

接下来 Q 行，每行描述一次操作，格式为 **1 i x** 或 **2 l r**：

- 对于所有第一类操作，保证 $1 \leq i \leq n$, $0 \leq x < pq$ ；
- 对于所有第二类操作，保证 $1 \leq l \leq r \leq n$ 。

Output

对于每一次 **2 l r** 操作，如果指定的卡片可以恰好被划分成若干个标准卡包，输出一行 **YES**；否则输出一行 **NO**。

Example

standard input	standard output
8 2 3 5	YES
0 0 2 3 4 1 4 2	NO
2 1 5	NO
2 6 8	YES
1 8 5	
2 6 8	
2 6 7	

Note

当 $p = 2, q = 3$ 时：

- 第一类标准卡包有两种，分别为： $\{0, 2, 4\}, \{1, 3, 5\}$ ；
- 第二类标准卡包有三种，分别为： $\{0, 3\}, \{1, 4\}, \{2, 5\}$ 。

对于第一次询问，区间内的卡片构成的多重集为 $\{0, 0, 2, 3, 4\}$ ，它可以被拆分为 $\{0, 2, 4\} + \{0, 3\}$ ，因此输出 **YES**。

对于最后一次询问，区间内的多重集为 $\{1, 4\}$ ，它恰好构成了一个第二类标准卡包，因此输出 **YES**。



Problem L. AC 自动机

给定确定性有限状态自动机 (DFA)，状态集 $Q = \{1, 2, \dots, n\}$ ，起始状态 $q = 1$ ，接受状态集 F 大小为 m ，字母表 $\Sigma = \{A, C\}$ 。计算能够被自动机接受且长度不短于 k 的字符串中字典序最小者。由于答案可能很长，因此如果答案的长度超过 n ，你只需要输出答案的最后 n 个字符。若满足前述要求的字符串不存在，输出 **WA**。

字符串被自动机接受，当且仅当存在一条从初始状态开始、到接受状态结束的 walk，使得走过的转移边的字母恰好形成了该字符串。walk 可以经过重复点、重复边。

Input

本题有多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 5 \times 10^5$)，表示测试数据组数。

对每组测试：

- 第一行包含用空格分隔的三个整数，依次表示：状态数 n ($2 \leq n \leq 10^6$)，接受状态数 m ($1 \leq m \leq n - 1$)，长度下限 k ($1 \leq k \leq 10^9$)。
- 第二行包含用空格分隔的 m 个整数，表示接受状态集。保证这些数均在 $[2, n]$ 之间，且不重复。
- 接下来 n 行，每行包含用空格分隔的 2 个整数，其中第 i 行的两个整数分别表示状态 i 沿 **A/C** 转移后的状态。

保证 $\sum n \leq 10^6$ 。

Output

输出 T 行，每行输出一个字符串，其中第 i 行表示第 i 组测试数据的答案。

Example

standard input	standard output
2	CCCCA
5 1 6	WA
4	
2 5	
5 3	
4 2	
5 5	
5 5	
3 1 2	
2	
2 3	
3 3	
3 3	

Note

样例第一组测试中，DFA 识别的语言为 $\{AC^{2t+1}A \mid t \geq 0\}$ 。



Problem M. 迭代不动点

给定三个整数 n, k, p 。
考虑所有函数

$$f : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}。$$

记 f^k 表示函数 f 的 k 次迭代。若 $x \in \{1, 2, \dots, n\}$ 满足

$$f^k(x) = x,$$

则称 x 是 f 的一个 k 阶迭代不动点。
求恰好有 p 个 k 阶迭代不动点的函数 f 的数量。答案对 $10^9 + 7$ 取模。

Input

本题有多组测试数据。
第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据组数。
接下来 T 行，每行包含三个整数 n, k, p ($1 \leq n, k \leq 10^6, 0 \leq p \leq n$)。
保证所有测试用例的 n 之和不超过 10^6 。

Output

对于每组测试数据，输出一行一个整数，表示答案对 $10^9 + 7$ 取模后的结果。

Example

standard input	standard output
7	12
3 2 2	1
2 1 0	8
3 1 0	2
3 2 0	3
3 3 3	10
4 2 4	54
4 1 2	