

Problem A. 树上的多米诺骨牌

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

给出一棵包含 n 个点，且以 1 号点为根的树。在第 i 号点上存在一个多米诺骨牌，也就是一个大小为 1×2 的长方形。这个多米诺骨牌被分为左右两个小正方形，分别包含 a_i 个点和 b_i 个点。这里的多米诺骨牌是没有方向的，这意味着你可以旋转这个多米诺骨牌，将左右两个正方形包含的点数交换。

有一群人生活在这棵树上，其中 x 号点的居民，可以前往 x 的子树并收集所有多米诺骨牌（包含 x 号点本身的多米诺骨牌）。居民们希望将这些多米诺骨牌组成尽可能少的"接龙"。你需要对所有的 $i \in [1, n]$ 求出，在 i 号点的居民利用其子树内所有多米诺骨牌能拼出的最少"接龙"数量。我们认为每个居民的操作是互相独立的，也就是在每个居民开始收集之前，树上的每个点都包含恰好一个多米诺骨牌。

这里的"接龙"指代一个多米诺骨牌的序列，其中多米诺骨牌按顺序横向放置，且保证前一个骨牌右侧的点数等于后一个骨牌左侧的点数。

Input

输入的第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)，代表树的点数。
接下来一行包含 n 个整数 a_i ($1 \leq a_i \leq n$)，表示骨牌左侧正方形的点数。
接下来一行包含 n 个整数 b_i ($1 \leq b_i \leq n$)，表示骨牌右侧正方形的点数。
接下来 $n - 1$ 行，每行包含两个整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示从 u 号点到 v 号点之间有一条边。保证给出的 $n - 1$ 条边能将 n 个点连成一棵树。

Output

输出一行 n 个整数，第 i 个整数代表 i 号点的居民可以拼出的最少"接龙"数量。

Example

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

Note

对于样例：

- 对于 2 号点的居民，一种可能的接龙方案为：[4|6][6|7] 和 [6|5]。需要注意，这一方案旋转了 2 号点的骨牌。
- 对于 3 号点的居民，一种可能的接龙方案为：[1|2][2|3] 和 [4|4]。
- 对于 1 号点的居民，一种可能的接龙方案为：[4|4][4|6][6|7][7|7]、[1|2][2|3] 和 [6|5]。

Problem B. 进制转换享受者

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

小 T 得到了 n 个数组 $A_{1..n}$, 第 i 个数组 A_i 的长度为 l_i 。另外, 小 T 给出了两个整数 y 和 M , 并希望你可以确定所有在 $[2, M]$ 内且满足如下要求的正整数 s 。

- 按顺序枚举 $i \in [1, n]$:
 - 将数组 A_i 看作 s 进制下的数字。如果数组 A_i 中存在一个不小于 s 的数字, 则认为不满足要求。
 - 对 A_i 进行进制转换, 将得到的结果 s' 赋值给 s 。
- 初始值满足要求, 当且仅当在完成上述 n 次进制转换后, $s = y$ 成立。

具体来说, 给定一个数组 $a_{1..k}$ 以及初始进制 s , 那么将数组 a 作为 s 进制下的数字进行转换, 得到的结果为:

$$s' = \sum_{i=1}^k a_i s^{k-i} = a_1 s^{k-1} + a_2 s^{k-2} + \cdots + a_k$$

小 T 的研究发现, 满足条件的正整数 s , 要么不存在, 要么恰好为某个区间 $[l, r]$ 内的全体正整数。你需要向小 T 汇报对应的 l, r , 或者告诉他不存在满足条件的正整数 s 。

Input

本题包含多组测试数据, 第一行包含一个整数 T ($1 \leq T \leq 10^5$), 代表测试数据数量。

对于每组测试数据:

第一行包含三个整数 n, y, M ($1 \leq n \leq 2 \times 10^5, 1 \leq y \leq 10^9, 2 \leq M \leq 10^9$)。

接下来的 n 行, 第 i 行首先包含一个整数 l_i ($1 \leq l_i \leq 2 \times 10^5$), 代表第 i 个数组 A_i 的长度。紧接着包含 l_i 个整数 $A_{i,j}$ ($0 \leq A_{i,j} \leq 10^9$), 代表数组 A_i 的内容。保证数组的第一个元素 $A_{i,1}$ 不为 0。

数据保证一次输入中所有数组的长度和不超过 2×10^5 。

Output

对于每组测试数据:

如果不存在满足要求的正整数 s , 输出一行 -1 -1;

否则, 假如所有满足要求的正整数形成区间 $[l, r]$, 则输出一行两个整数, 分别为 l 和 r 。

Example

standard input	standard output
3	5 5
4 39244 100	257 1000000000
2 1 1	-1 -1
2 1 1	
2 1 1	
6 1 1 4 5 1 4	
3 1000000000 1000000000	
3 1 0 0	
1 65536	
2 15258 51712	
2 27 3	
4 1 0 0 0	
1 27	

Problem C. 栈

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

对于一个排列 P ，定义 $f(P)$ 如下：

```
function f(P):  
    stack = []  
  
    for element in P:  
        while stack is not empty and stack.top() > element:  
            stack.pop()  
        stack.push(element)  
  
    return the size of stack
```

给定一个整数 n ，求所有长度为 n 的排列 P' 的 $(f(P'))^3$ 的和，对 998244353 取模。

Input

本题有多组输入数据。
第一行输入一个正整数 $T(1 \leq T \leq 10^5)$ ，表示输入数据组数。
接下来的每组输入数据，输入一个正整数 $n(1 \leq n \leq 5 \cdot 10^5)$ 。

Output

对于每组输入数据，输出一行一个整数，表示答案。

Example

standard input	standard output
2	53
3	805156151
3741	

Problem D. 漂亮矩阵

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

一个漂亮矩阵 A 具有 n 行和 n 列，满足以下条件：

- 所有元素都是介于 0 和 m 之间的整数。
- 对于 $1 \leq i \leq n$ ，有 $A_{i,1} = 0$ 。
- 对于 $1 \leq i \leq n$ 和 $1 \leq j < n$ ，有 $A_{i,j} \leq A_{i,j+1}$ 。
- 对于每一个整数四元组 (i,j,k,l) ，如果满足 $1 \leq i < k \leq n$ 和 $1 \leq j < l \leq n$ ，则有 $A_{i,j} + A_{k,l} \leq A_{i,l} + A_{k,j}$ 。

给定两个整数 n 和 m ，求漂亮矩阵的数量，结果对 998244353 取模。

Input

第一行包含两个整数 n 和 m ($2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$)。

Output

输出答案。

Examples

standard input	standard output
2 1	3
29912 86488	461289656
4 3	455

Problem E. 数组

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

给定一个长度为 n ，初始元素全为 0 的数组 a ，以及一个长度为 n 的数组 p 。你需要处理 q 个操作，操作分为两种类型：

- 1. 更新操作：给定四个整数 $opt = 1, l, r, x$ ，对于所有 $l \leq i \leq r$ ，将元素 a_i 加上 x 。
- 2. 查询操作：给定三个整数 $opt = 2, l, r$ ，表示计算 $\sum_{i=l}^r a_{p_i}$ 。

对于每个查询操作，输出查询结果。本题强制在线，每次操作输入的参数 l, r, x 要异或上一次输出的答案（第一次询问操作前不异或）。

Input

第一行包含两个整数 n 和 q ($1 \leq n, q \leq 10^5$)，分别表示数组 a 的长度和操作的数量。
第二行包含 n 个整数，表示数组 p ($1 \leq p_i \leq n$)。
接下来 q 行，每行描述一个操作，格式如上述两种类型之一。对于所有的解密后的输入，有 $1 \leq opt \leq 2, 1 \leq l \leq r \leq n$ 且 $1 \leq x \leq 10^8$ 。

Output

对于每个查询操作，输出一行，每行一个整数，表示查询的结果。

Example

standard input	standard output
6 10	34
3 6 5 1 2 6	38
1 1 3 8	81
1 2 3 9	
1 2 5 4	
1 2 3 9	
2 1 3	
1 33 33 38	
1 32 32 35	
2 35 33	
1 36 34 36	
2 39 35	

Problem F. 随机线段树

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 256 megabytes

定义一个随机线段树为一个二叉树，其中每个节点表示一个闭区间 $[l, r]$:

- 如果 $l = r$ ，则该节点为叶子节点。
- 如果 $l < r$ 且区间的长度 $(r - l + 1)$ 是偶数：设 $x = \lfloor (l + r)/2 \rfloor$ 。节点的左子节点表示 $[l, x]$ ，右子节点表示 $[x + 1, r]$ 。
- 如果 $l < r$ 且区间的长度 $(r - l + 1)$ 是奇数：设 $x = (l + r)/2$ 。以 $1/2$ 的概率，左子节点表示 $[l, x]$ ，右子节点表示 $[x + 1, r]$ ；以 $1/2$ 的概率，左子节点表示 $[l, x - 1]$ ，右子节点表示 $[x, r]$ 。

给定一个线段树，定义 $\text{cost}(x, y)$ 如下：

在查询区间 $[x, y]$ 时，从根节点开始，设当前在节点 $[l, r]$ ，过程如下：

- 如果 $[x, y]$ 包含 $[l, r]$ ，查询结束。
- 否则，如果左儿子节点与 $[x, y]$ 相交，则查询左儿子节点；如果右儿子节点与 $[x, y]$ 相交，则查询右儿子节点。
- $\text{cost}(x, y)$ 的值是查询过程中访问的节点数量。

给定 n ，考虑一个随机线段树，其根节点表示 $[1, n]$ 。对于 $1 \leq i \leq 2 \cdot n$ ，计算 $\text{cost}(x, y)$ 等于 i 的区间 $[x, y]$ 的期望数量，结果对 998244353 取模。

Input

第一行包含两个整数 n ($1 \leq n \leq 10^5$)。

Output

输出 $2 \cdot n$ 行。每行包含一个整数，表示答案对 998244353 取模。

Examples

standard input	standard output
3	1 2 2 1 0 0
4	1 2 4 2 1 0 0 0
5	1 2 4 4 499122179 1 499122177 0 0 0

Problem G. 转身

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

在一条数轴上有 n 个人，每个人面朝左（'L'）或右（'R'）。给定一个长度为 n 的字符串 S ，其中每个字符是 'L' 或 'R'，表示每个人的初始方向。在每个时刻，当两个相邻的人面对面时（即，左边的人面朝右，右边的人面朝左），他们会同时转身：如果一个人面朝左，他会转向右；反之亦然。

你的任务是确定在没有人再改变方向之后所需的时间。

此外，还有 q 次修改。每次修改给出一个索引 x ，意味着第 x 个人的初始方向被翻转（从 'L' 变为 'R' 或反之）。在每次修改后，输出上述问题的答案。

Input

第一行包含两个整数 n ($1 \leq n \leq 2 \cdot 10^5$) 和 q ($1 \leq q \leq 2 \cdot 10^5$)。

第二行包含一个长度为 n 的字符串 S ，表示每个人的初始方向。

接下来的 q 行包含一个整数 x ($1 \leq x \leq n$)。

Output

输出 q 行。每行包含一个整数，表示答案。

Examples

standard input	standard output
5 5 LRLRL 5 1 3 4 3	1 2 0 3 3
5 5 RLLLL 1 2 3 4 5	0 3 3 3 0

Problem H. 背包问题

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

有 n 种物品，每种物品个数无限，有体积 a_i 和价值 b_i 两种属性。
定义 $f(v)$ 为：你有一个体积为 v 的背包，要求选若干个物品放入背包中，使物品的总价值最大，且背包中物品的总体积不超过背包的容量。
给定这 n 种物品和一个数 V ，计算 $\sum_{v=1}^V f(v) \bmod 998244353$ 。

Input

第一行包含两个正整数 n ($1 \leq n \leq 10^5$)，和 V ($1 \leq V \leq 10^{18}$)。
接下来 n 行，每行包含两个正整数 a_i ($1 \leq a_i \leq 1500$)，和 b_i ($1 \leq b_i \leq 10^9$)，表示每种物品的属性。

Output

输出一个数，表示答案。

Examples

standard input	standard output
5 10 7 9 5 3 1 1 2 4 9 10	105
10 20 4 11 10 13 6 11 10 1 5 8 9 13 7 7 13 8 3 13 7 6	819

Problem I. 最长公共子串

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

两个字符串之间的最长公共子串是出现在两个字符串中的最长连续字符序列。
给定一个长度为 n 的字符串 s 和 q 次查询，每次查询给出两个参数 l 和 r ，计算前缀 $s[1..l]$ 和后缀 $s[r..n]$ 之间的最长公共子串的长度。

Input

第一行包含两个正整数 n 和 q ($1 \leq n, q \leq 2 \cdot 10^5$) — 字符串 s 的长度和查询操作的数量。
第二行包含一个由 n 个小写英文字母组成的字符串 s 。
接下来的 q 行，每行包含两个正整数 l 和 r ($1 \leq l, r \leq n$) 。

Output

对于每个查询操作，在单独的一行上输出一个整数，表示 $s[1..l]$ 和 $s[r..n]$ 之间的最长公共子串的长度。

Example

standard input	standard output
10 14	1
aaabaaaaab	2
1 2	3
2 3	4
3 4	4
4 5	4
5 6	3
6 7	2
7 8	1
8 9	7
9 10	3
7 1	1
4 8	4
6 10	2
5 5	
2 5	

Problem J. 树上游戏

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

小 Q 正在游玩一个树上游戏。

游戏的规则如下：每个回合，小 Q 可以选择树上一个未被删除的点，将其以及其所有的直接连边删除。当所有边都被删除时，游戏结束（注意，节点不必全部都被删除）。

现在小 Q 想要知道，如果每一回合**均匀随机**地选取一个可以删除的节点，游戏期望进行多少个回合后结束？输出答案对 998244353 取模后的结果。

Input

第一行输入一个正整数 $n(2 \leq n \leq 5000)$ ，表示树的节点数。

接下来 $n - 1$ 行，每行两个正整数 $x, y(1 \leq x, y \leq n)$ ，表示 x 到 y 之间有一条边。

Output

输出一行一个整数，表示答案。

Examples

standard input	standard output
3 1 2 2 3	665496237
5 1 2 2 3 3 4 2 5	598946615

Problem K. 最大gcd

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

给定长为 n 的序列 a ，你需要执行恰好一次操作：选择一个区间，以及一个非负整数 X ，将区间中的所有数加上 X ，最大化序列中所有数的 gcd 。输出所有数的 gcd 的最大值，若为无穷大，输出 0。

Input

本题有多组输入数据。
第一行输入一个正整数 $T(1 \leq T \leq 10^5)$ ，表示输入数据组数。
接下来的每组输入数据，首先输入一个正整数 $n(1 \leq n \leq 10^5)$ ，表示序列的长度。
接下来输入 n 个正整数，表示序列 $a(1 \leq a_i \leq 10^5)$ 。
保证 $\sum n \leq 2 \cdot 10^5$ 。

Output

对于每组输入数据，输出一行一个整数，表示答案。

Example

standard input	standard output
5	514
2	0
114 514	1
1	4
2	3
5	
1 2 3 5 8	
5	
4 3 3 3 4	
5	
6 1 4 7 9	

Problem L. 最小括号串

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

给定 m 个区间，求长为 $2n$ 的字典序最小的合法括号串，满足每个区间内至少有一个左括号。若无解，则输出 -1。

我们规定左括号的字典序小于右括号。

Input

本题包含多组测试数据，第一行包含一个整数 T ($1 \leq T \leq 10^5$)，代表测试数据数量。
对于每组输入数据，第一行先输入两个正整数 n, m ($1 \leq n, m \leq 10^5$)，分别表示需要构造的括号串的长度的一半，以及限制区间的个数。
接下来 m 行，每行两个正整数 l, r ($1 \leq l \leq r \leq 2 \cdot n$)，表示构造的括号串需要满足 $[l, r]$ 中至少有一个左括号。
保证 $\sum n \leq 2 \cdot 10^5$ ，且 $\sum m \leq 4 \cdot 10^5$ 。

Output

对于每组输入数据，输出一个字符串，表示构造的字典序最小的合法括号串。若无解，输出 -1。

Example

standard input	standard output
3	()()
2 2	-1
1 2	(((()))())
3 4	
3 4	
1 1	
2 2	
3 3	
4 4	
5 3	
2 4	
3 9	
7 10	

Problem M. 最小差值

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

给定 n 个**不重复**的整数 a_i ，你需要将这些数位恰好分为两个集合，各自任意排列构造出两个 m 进制数 A 、 B （允许前导0的出现），最小化 A 和 B 的差值。

Input

本题有多组输入数据。第一行包含一个正整数 T ($1 \leq T \leq 10^5$)，代表测试数据组数。
对于每组输入数据，第一行输入两个正整数 n, m ($2 \leq n, m \leq 10^5$)，分别表示整数的个数和进制。
接下来一行输入 n 个整数 a_i ($0 \leq a_i < m$)，保证这 n 个整数**互不相同**。
保证 $\sum n \leq 2 \cdot 10^5$ 。

Output

因为最小差值可能很大，你只需输出差值转为十进制后 $\text{mod } 998244353$ 的结果。
对于每一个输入数据，输出一行一个整数，表示结果。

Example

standard input	standard output
3	7
4 10	469
3 4 5 6	124
7 10	
1 2 3 4 5 6 7	
6 16	
2 3 5 7 11 13	