

北京理工大学第十七届程序设计大赛 - 决赛

The 17-th BIT Campus Programming Contest - Final



题目列表

Problem List

A	赛前须知
B	小 L 的围棋
C	小 L 的旅行
D	小 L 的数学题
E	MyGO!!!!
F	Harmini
G	小 P 玩钢 4
H	LCA Determinant
I	星光指引前路
J	pass
K	str 进制
L	Equation
M	小 H 的糖果

请勿在比赛开始前翻阅试题!

Do not open before the contest starts!

2023 年 11 月 26 日

Problem A. 赛前须知

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

请注意，这是一道编程题。

亲爱的选手们，

祝贺大家在网络赛中取得优异的成绩，晋级北京理工大学第十七届程序设计大赛决赛。在此，我谨代表竞赛裁判组，向各位同学们，表示热烈的欢迎和衷心的祝贺。

本次比赛由计算机学院与信创学院联合主办。信创学院是北京理工大学与中软国际携手打造的创新学院，得到了中软国际、深开鸿、华为等企业的鼎力支持。首届“开源鸿蒙英才班”在中关村校区举办开班典礼，标志着信创学院在开源鸿蒙领域的深度合作迈出了关键一步。学院致力于培养软件领域的新一代领军领导人才，强调“延安根、军工魂、领军人”的红色基因，致力于推进国家软件产业创新发展。这一合作旨在共同打造关键基础软件领域的“国之重器”，引领中国在全球软件领域的崛起。开源鸿蒙 (OpenHarmony) 是由开放原子开源基金会 (OpenAtom Foundation) 孵化及运营的开源项目，目标是面向全场景、全连接、全智能时代、基于开源的方式，搭建一个智能终端设备操作系统的框架和平台，促进万物互联产业的繁荣发展。

现场赛的规则与网络赛有所不同，请仔细阅读以下注意事项：

1. 请遵守赛场纪律，勿在比赛开始之前拆封试题袋、触碰鼠标或者键盘！比赛开始后，禁止使用自己的电子设备！比赛过程中，禁止与队友和工作人员以外的任何人以任何形式交流。
2. 本次比赛采用 ICPC 赛制（也称为 ACM 赛制）。ICPC 赛制以各个队伍的数量为第一关键字降序进行排序，当过题数量相同时，以罚时作为第二关键字升序进行排序，罚时的计算方式是：已通过的题目的 AC 的时间的总和 + 这些题目提交前错误的次数（不含编译错误）*20 分钟。
3. 本次比赛允许的竞赛语言包括：C, C++, Java, Python。
4. 本次比赛**不允许**与除比赛服务器等完成比赛必要的计算机以外的任何计算机进行任何形式的网络通信，但可以查阅本队在赛前准备的任何**纸质**材料。
5. 本次比赛的题目**不按难度顺序排序**，当遇到不会做的题目时，请继续阅读下一道题目。
6. 如果对题目有疑问，应当通过比赛系统的“clarification”功能向裁判组提出。

为了确保各位同学都已经阅读并理解上述内容，请大家完成以下 10 道**单选题**：

1. 以下行为中，被禁止的**不包括**（ ）
 - A. 在比赛开始前，和队友打情骂俏
 - B. 在比赛开始前，拆封试题袋
 - C. 在比赛开始前，操作比赛用机
 - D. 在比赛开始后，使用自己携带的计算器

2. 作为参赛选手，你可以在比赛中（ ）
- A. 使用手机玩原神
 - B. 用言语嘲讽其他队伍的队员
 - C. 查阅提前书写在餐巾纸中的小抄
 - D. 使用 SSH 登录其他队伍的比赛机器
3. 本次比赛的赛制是（ ）
- A. ICPC 赛制
 - B. LeetCode 赛制
 - C. IOI 赛制
 - D. 乐学赛制
4. 在 ICPC 赛制中，下列四个选项中，成绩排名第一的是（ ）
- A. 通过 1 题，罚时 114514 分钟
 - B. 通过 1 题，罚时 1919810 分钟
 - C. 通过 13 题，罚时 114514 分钟
 - D. 通过 13 题，罚时 1919810 分钟
5. 以下是某个参赛队的比赛记录，在 ICPC 赛制中，该队此时的罚时是（ ）分钟

提交时间（分钟）	题目编号	结果
10	B	WRONG-ANSWER
20	A	CORRECT
30	B	COMPILER-ERROR
40	B	CORRECT
50	B	WRONG-ANSWER
60	C	CORRECT

- A. 140
 - B. 160
 - C. 180
 - D. 210
6. 在本次比赛中，**不能**使用以下哪种语言（ ）
- A. C
 - B. C++
 - C. JavaScript
 - D. Python
7. 当你对题目有疑问时，可以（ ）
- A. 使用比赛系统的“clarification”功能向裁判组提出
 - B. 询问旁边队伍的参赛选手
 - C. 在赛场大声问候出题人
 - D. 躺在赛场的地板上，大声哭闹以吸引出题人的注意
8. 在比赛中，当你不会做某道题目时，你可以（ ）
- A. 询问附近的队伍该题的解法
 - B. 询问 ChatGPT 该题的解法
 - C. 放弃求解该题目，尝试解出其他题目
 - D. 通过“clarification”功能向裁判询问该题的解法

9. 本次比赛由计算机学院与（ ）联合主办

- A. 信创学院
- B. 原神学院
- C. 魏公村职业汽修学院
- D. 电竞学院

10. “鸿蒙”的英文名为（ ）

- A. Harmini
- B. Howmoney
- C. Harmony
- D. Homony

Input

本题不需要读取输入数据。

Output

输出共 10 行。对于每道题目，输出一行一个大写字母，表示你选择的选项。

Example

standard output	
D	
C	
B	
A	
A	
A	
A	
B	
C	
D	

Note

样例输出仅作为格式参考，不代表题目的正确答案。

Problem B. 小 L 的围棋

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

围棋是我国古代的智慧结晶。

最近经常上网的小 L 迷上了看一位围棋八冠王选手的直播，为了去了解他的更多，小 L 便想要去学习围棋，希望有朝一日能与他博弈。但是奈何小 L 的围棋天赋不高，但是在学习围棋的过程中，他想到了一个类似于围棋的新游戏：

棋盘大小为 $n * n$ ，左上角坐标为 $(1, 1)$ ，从上到下 x 坐标递增，从左至右 y 坐标递增，两位玩家在棋盘右上半部分（即 $x \leq y$ 的部分）轮流落黑白色的棋子（先落黑子再落白子），一颗棋子的左下能落子的区域（包括该行该列）称为该棋子的“控制区”，保证棋盘上所有的棋子的“控制区”只会真包含或者不交。

一颗棋子的“气”定义为它“控制区”内权值不小于它的同色棋子“气”的最大值 +1，如果“控制区”内除自己以外没有同色棋子则该点“气”为 1，若某区域内没有 x 色棋子，则该区域内 x 色棋子的“气”的最大值视为 0。

同时，小 L 有一个序列 a ，并定义格点 $M(x, y)$ 的权值为 $\min_{x \leq i \leq y} (a_i)$ 。

整局棋下完后分别算双方的积分，积分规则如下：

I. 一颗棋子的“控制区”内与该颗棋子颜色相同（包括该颗棋子）的所有棋子所在的格点中，相同权值出现的最大次数如果大于“控制区”内另一种颜色的棋子所在格点的相同权值出现的最大次数，则称该方玩家获得一分。

II. 一颗棋子的“气”若大于其“控制区”中所有异色棋子的“气”，则该方玩家获得一分。

现在给定序列 a 和一次游戏两人轮流落子的坐标，求两人最后的分数分别为多少。

Input

第一行输入两个正整数 n, m ($1 \leq n, m \leq 2 \times 10^5$)，代表序列 a 的长度和双方所下棋子的个数。

第二行输入 n 个数，代表序列 a ($0 \leq a_i \leq 10^9$) 的值。

接下来 m 行每行输入两个数 x, y ($1 \leq x \leq y \leq n$)，代表每个棋子的横纵坐标。

数据保证棋盘上所有的棋子的“控制区”只会真包含或者不交。

Output

一行两个整数，代表黑棋方的分数和白棋方的分数，中间用一个空格隔开。

Examples

standard input	standard output
5 4 1 2 3 4 1 1 5 1 3 1 1 4 5	3 2
13 9 1 3 5 6 4 2 9 21 10 6 21 1 3 1 13 11 12 5 8 6 7 1 9 1 4 3 4 1 2 10 13	3 8

Note

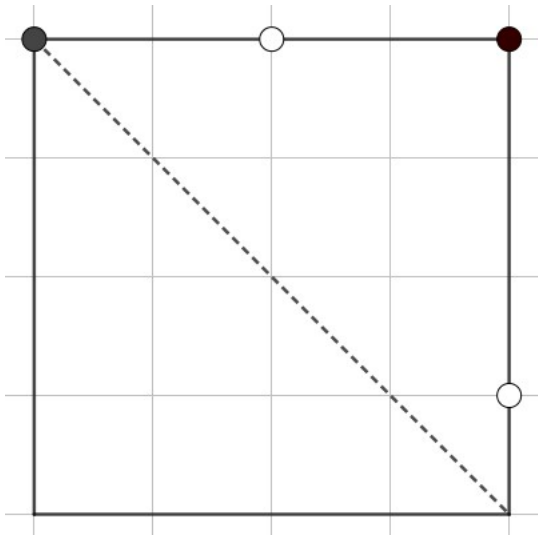


Fig. 1. 样例一示意图

在样例 1 中，四个有棋子的格点的权值均为 1,[1,1],[1,3],[4,5] 的控制区内均无同色棋子，则它们的气均为 1,[1,5] 控制区内有 [1,1] 这颗同色棋子，则 [1,5] 的气为 2。

对于规则 I:

[1,5] 控制区内黑子出现次数最多的数字为 1，为 2 次，白子出现次数最多的数字为 1，为 2 次，不计分；

[1,3] 控制区内白子出现次数最多的数字为 1，为 1 次，黑子出现次数最多的数字为 1，为 1 次，不计分；

[1,1] 控制区内黑子出现次数最多的数字为 1，为 1 次，没有白子出现，黑子方计一分；

[4,5] 控制区内白子出现次数最多的数字为 1，为 1 次，没有黑子出现，白子方计一分；

对于规则 II：

[1,5] 控制区内白子最大的气为 1，而其本身的气为 2，黑子方计一分；

[1,3] 控制区内黑子最大的气为 1，而其本身的气为 1，不计分；

[1,1] 控制区内白子最大的气为 0，而其本身的气为 1，黑子方计一分；

[4,5] 控制区内黑子最大的气为 0，而其本身的气为 1，白子方计一分；

综上所述，黑子得分为 3，白子得分为 2。

Problem C. 小 L 的旅行

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 512 megabytes

今天是周末！小 L 决定去北京城区玩。

根据小 L 的经验，北京城区有 n 个景点，一共有 m 条单向边将这些景点连接起来，第 i 条边从第 u_i 出发到达第 v_i 个景点，经过一条边需要耗费 1 分钟的时间。而小 L 此时在第 1 个景点。

小 L 觉得这样出行太慢啦！所以小 L 找住在房山区的小 P 要了一张《北京传送门地图》。具体来说，第 i 个景点具有 a_i 的魔力值，如果存在一个景点 j 满足 a_i and $a_j = a_j$ (and 指按位与运算，详见本题的 Notes 部分)，那么就存在一个从第 i 个景点通向第 j 个景点的**单向**传送门，她就可以花费 1 分钟的时间，直接从 i 到达 j 。

如此庞大的交通网络让小 L 看得眼花缭乱，而小 P 这周末去合肥旅游了，所以她决定直接来问你：如果她决定去第 i 个景点，最少需要花费多少分钟？你需要对于所有 $i \in [1, n]$ 都回答一遍。

Input

第一行输入两个整数 n, m ($1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 3 \times 10^5$)，代表景点数量和边的数量。

第二行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 2^{20}$)，代表各个景点的魔力值。

接下来的 m 行，每行输入两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，代表第 i 条边是从第 u_i 个到第 v_i 个景点的单向边。

Output

输出 n 行，第 i 行表示从第 1 个景点到达第 i 个景点最少需要多少分钟。如果到达不了第 i 个节点，则第 i 行输出 -1 。

Example

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

Note

按位与运算：

对于两个二进制的按位与运算，只有当他们同一位的数字同为 1 时才为 1，否则取 0。相当于数学与的同真为真，在 C/C++ 中对应的运算符是 `&`。例如： $(1010)_2$ and $(1100)_2 = (1000)_2$ 。

Problem D. 小 L 的数学题

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

小 L 想让小 P 快点从合肥回来陪她，但是小 P 想在合肥多玩一会，怎么办呢？于是小 P 给小 L 出了一道数学题，并且承诺她当她算出来的时候就会来到小 L 的身边。

这个数学题是这样的：对于一个正整数 n 而言，如果存在 n 的一个因子 d ，满足 d 与 n/d 互质，则称 d 是 n 的好因子。设 $f(n)$ 表示 n 的好因子的个数，例如 6 的好因子有：1, 2, 3, 6，则 $f(6) = 4$ 。现在要求：

$$\sum_{i=1}^n f(i)$$

小 L 为此很苦恼，她想快点让小 P 从合肥回来，但是她又根本不会算这题。所以她找到正在打比赛的你，想请你帮忙算一下这个题的答案。

Input

输入一个正整数 n ($1 \leq n \leq 10^{12}$)。

Output

输出一个数，为答案对 998244353 取模的值。

Example

standard input	standard output
10	23

Problem E. MyGO!!!!

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

小 A 最近喜欢上了一部番《BanG Dream! It's MyGO!!!!》。



Fig. 2. MyGO

现在他有一个长度为 n 的数列 $\{a_n\}$ ，现在要把它分成若干段，要求每段的异或和均严格大于 k 。

划分出的每段必须连续，并且每个元素恰好在其中的一个段中。

小 A 想知道，对于所有的划分方案，划分段数三次方的总和是多少，由于这个数可能很大，你只需要输出它对 998244353 取模后的结果即可。

形式化的，对于所有数列 $\{a_n\}$ 的划分 $\{x_0, x_1, x_2 \dots x_m\}$ ，满足 $x_0 = 0$, $x_m = n$ 和 $x_{i-1} < x_i$ ，求：

$$\sum_{x_0, x_1, x_2 \dots x_m} m^3 \left[\bigwedge_{i=0}^{m-1} \left[\bigoplus_{j=x_i+1}^{x_{i+1}} a_j > k \right] \right] \mod 998244353$$

其中，部分符号解释如下：

- $[x]$ ：若 x 是一个条件表达式，若 x 成立，则 $[x]$ 的值为 1，否则 x 的值为 0。
- $\bigwedge$ ：表示其后所有条件逻辑与的结果，即该条件成立当且仅当其后所有条件成立。
- $\bigoplus$ ：表示其后所有数值按位异或的结果。
- 异或：对于两个二进制数的按位异或运算，只有当他们同一位的数值相同时才为 1，否则取 0。在 C/C++ 中对应的运算符是 $\wedge$ 。例如： $(1010)_2 \text{ xor } (1100)_2 = (0110)_2$ 。

Input

输入共包含两行。

第一行包含两个整数 n, k ($1 \leq n \leq 10^6, 1 \leq k \leq 10^9$)。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^6$)。

Output

输出一个数 n , 表示所有方案中划分段数三次方的和对 998244353 取模后的结果。

Examples

standard input	standard output
3 2 1 2 3	8
4 4 1 4 7 9	36

Problem F. Harmini

Input file: standard input
Output file: standard output
Time limit: 8 seconds
Memory limit: 256 megabytes

这是一道交互题。

鸿蒙系统是一款全新的面向全场景的分布式操作系统，创造一个超级虚拟终端互联的世界，将人、设备、场景有机地联系在一起，将消费者在全场景生活中接触的多种智能终端，实现极速发现、极速连接、硬件互助、资源共享，用合适的设备提供场景体验。

你也想实现一个可以简易的连接多种终端的系统，称为 *Harmini*。在 *Harmini* 系统中，每个终端被视为一个节点，节点之间的连接被表示为一条边。所有节点相互之间都应该可以通过边连接，但由于技术力的原因，你希望边数尽可能的少。换句话说，***Harmini* 系统可以被抽象为一棵树**。

Harmini 中的每条边都被分配了一个整数权值，当两个终端 u, v 进行通讯时，通讯的权值为 u, v 之间简单路径上的所有边的权值的二进制异或和。

不幸的是，粗心的你将每条边的权值都忘记了。你只能通过尝试通讯两个节点，来获得它们之间的简单路径上的边的权值异或和。出于 *Harmini* 的某个 bug，只有距离恰好为 k 的点之间可以进行通讯。

假设 *Harmini* 中一共有 n 个点，你需要在 n 次询问内获取所有边的权值。

Interaction Protocol

第一行首先输入两个整数 n, k ($2 \leq k < n \leq 5 \times 10^4$)，分别代表树的节点数和询问的距离限制。

接下来 $n - 1$ 行，第 i 行输入两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，代表第 i 条边连接了节点 u_i, v_i 。保证输入为一棵树。

你最多可以进行 n 次询问，询问的格式为 “? $u\ v$ ”，表示询问 u, v 之间的简单路径的边权异或和。你需要保证 u, v 之间的简单路径的边数恰好为 k 。根据你的询问，系统将会输出一个整数 x ($0 \leq x < 2^{31}$)，代表询问的结果。

当你计算出结果后，输出一行 “! $w_1\ w_2 \cdots w_{n-1}$ ”，其中 w_i 代表输入的第 i 条边的权值，权值为 $[0, 2^{31})$ 之间的整数，然后立刻停止程序。

如果没有在 n 次询问内得到结果的方法，则你可以在任何时间输出一行 “! No solution”。

如果你的询问次数超过 n ，或询问的节点之间简单路径的边数不为 k ，你将会得到 WRONG-ANSWER。

为了让系统接收到你的询问，你需要在**输出任何询问后立刻清空输出缓存**，否则你将得到 TIMELIMIT 的结果。清空输出缓存可以用以下方法：

- 在 C++ 中使用 `fflush(stdout)` 或 `cout.flush()`。
- 在 Java 中使用 `System.out.flush()`。
- 在 Python 中使用 `stdout.flush()`。

Example

standard input	standard output
8 3	
1 2	
2 3	
3 4	
4 5	
5 6	
7 3	
8 4	
	? 3 6
0	
	? 2 5
1	
	? 1 4
0	
	? 7 5
1	
	? 8 6
0	
	? 8 2
0	
	? 7 1
1	
	! 1 0 1 0 1 0 1

Problem G. 小 P 玩钢 4

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

小 P 在合肥玩了一天，晚上回到酒店后决定开一把钢铁雄心 4。钢铁雄心 4 是一款二战模拟器，玩家可以操纵部队进攻敌国。玩了没多久，小 P 很快便包围了敌军的部队。

具体来说，当前这一片战场可以看成是一块 $n \times m$ 的网格，每个格子上有一个数字，第 i 行第 j 列的数字为 $a_{i,j}$ 。如果 $a_{i,j} > 0$ ，则代表这块格子上有数量为 $a_{i,j}$ 的敌军部队，如果 $a_{i,j} = 0$ ，则代表这块格子是我军部队。**在这片战场之外也全是我军部队。**

显然，这些敌军部队深陷我军的包围之中，我们需要快速消灭包围圈中的敌人。我们每次可以选择一块与我军部队相邻的有敌军的格子（假设是第 i 行第 j 列），然后一共需要消耗 $a_{i,j}$ 加上这个格子四周的敌军部队数量之和个单位的装备消灭掉这个格子上的敌人，随后我军部队进入这个格子， $a_{i,j}$ 变为 0。

由于战场宽度的限制，导致每个格子最多只有一个敌军部队，即 $a_{i,j} \in \{0,1\}$ 。现在小 P 想要知道，最少消耗多少装备能够消灭掉所有敌军部队？

Input

第一行输入两个整数 n, m ($1 \leq n, m \leq 1000$)，表示战场的行数和列数。
接下来 n 行，每行输入 m 个整数，第 i 行第 j 个数为 $a_{i,j}$ ($a_{i,j} \in \{0,1\}$)。

Output

输出一行，即最少消耗的装备数量。

Example

standard input	standard output
5 5 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0	9

Note

在样例中，我们可以先攻击 $a_{3,3}$ 的部队，需要消耗 $1 + 3 = 4$ 个单位的装备。接下来攻击 $a_{3,4}$ 的部队，需要消耗 $1 + 1 = 2$ 个单位的装备。接下来三个剩下的敌军部队任选一个顺序攻击即可。因此总的花费为 9。可以证明不存在更优的方案。

Problem H. LCA Determinant

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

You are given a tree consisting of n vertices. Vertices are denoted by $1, 2, \dots, n$, while vertex 1 is the root of the tree. Each vertices has a weight a_i .

Consider a matrix

$$A := \left(a_{\text{lca}(i,j)} \right)_{n \times n} = \begin{pmatrix} a_{\text{lca}(1,1)} & a_{\text{lca}(1,2)} & \cdots & a_{\text{lca}(1,n)} \\ a_{\text{lca}(2,1)} & a_{\text{lca}(2,2)} & \cdots & a_{\text{lca}(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\text{lca}(n,1)} & a_{\text{lca}(n,2)} & \cdots & a_{\text{lca}(n,n)} \end{pmatrix}$$

where $\text{lca}(i, j)$ denote the lowest common ancestor of vertex i and vertex j . Please calculate the determinant of matrix A . Since the answer could be large, output your answer modulo 998244353.

If you are unfamiliar with graph theory, see notes for clarification of terms used above.

Input

The input consists of three lines.

The first line consists of only one integer n ($2 \leq n \leq 5 \times 10^5$), the size of the tree.

The second line consists of $n - 1$ integers $p_2, p_3, \dots, p_n$ ($1 \leq p_i < i$), where p_i denotes the parent of vertex i .

The last line consists of n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 998244353$), the weight of each vertex.

It is guaranteed that the input data forms a valid rooted tree.

Since the input amount is large, using efficient reading method is suggested. In C++ you may close the synchronization option of `cin/cout` via following code.

```
ios::sync_with_stdio(0);  
cin.tie(0);cout.tie(0);
```

Output

A single integer, $\det A \bmod 998244353$.

Examples

standard input
3 1 2 5 1 4
standard output
998244293

standard input
5 1 1 2 2 1 2 3 4 5
standard output
12

standard input
5 1 1 3 4 20231126 514514514 19260817 353442899 112358111
standard output
86351415

Note

In the first example,

$$\det A = \det \begin{pmatrix} 5 & 5 & 5 \\ 5 & 1 & 1 \\ 5 & 1 & 4 \end{pmatrix} = -60$$

In the second example,

$$\det A = \det \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 & 2 \\ 1 & 1 & 3 & 1 & 1 \\ 1 & 2 & 1 & 4 & 2 \\ 1 & 2 & 1 & 2 & 5 \end{pmatrix} = 12$$

Terms used above:

- A tree is a connected acyclic undirected graph. A rooted tree is a tree in which one vertex has been designated the root.
- In a rooted tree, the parent of a vertex v is the vertex connected to v on the path to the root; every vertex (except the root) has exactly one parent.
- The lowest common ancestor (LCA) of two vertices v and w in a tree is the lowest (i.e. deepest) node that has both v and w as descendants.

Problem I. 星光指引前路

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 512 megabytes

卡丘世界是一个二维平面，星绘在战地中放置了 n 个星空之门，星空之门是一个矩形，能提供守护星芒获取护甲值。

形式化地，每个星空之门可以用 (x_1, y_1, x_2, y_2, w) 表示，代表一个左上角坐标为 (x_1, y_1) ，右下角坐标为 (x_2, y_2) 的矩形，提供 w 点护甲值。

战地中有 m 位超弦体，第 i 名超弦体位于 (x_i, y_i) 的位置上，获取战场信息是非常重要的事情，所以超弦体 i 好奇涵盖她的所有星空之门中提供第 k_i 小的护甲值是多少？

一个星空之门 (x_1, y_1, x_2, y_2) 能够涵盖一个超弦体 (x, y) ，当且仅当 $x_1 \leq x \leq x_2$ 且 $y_1 \leq y \leq y_2$ 。

Input

第一行一个正整数 n ($n \leq 5 \times 10^4$) 代表星空之门的个数。

第二行到第 $n + 1$ 行，第 i 行五个整数 $x_{1i}, y_{1i}, x_{2i}, y_{2i}, w_i$ ($|x_1|, |y_1|, |x_2|, |y_2| \leq 10^9, 0 < w \leq 10^9, x_1 \leq x_2, y_1 \leq y_2$)，含义如题所述。

第 $n + 2$ 行一个正整数 m ($m \leq 10^5$) 代表超弦体数目。

接下来 m 行，每行三个整数 x_i, y_i, k_i ，意义如题所述， $|x|, |y| \leq 10^9, 1 \leq k \leq n$ 。

Output

输出共 m 行，第 i 行代表第 i 名超弦体想知道的信息。特别地，若涵盖其的星空之门数目不足 k_i ，输出 -1 。

Example

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

Problem J. pass

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

smzzl 驾驶着小车正在环海公路漫游，突然遇到了一个很窄的地方，还好他技术高超，他打算使用 b 站刚看到的单边桥方法通过这段路（你别管怎么上的墙）。由于 smzzl 喜欢漂移，又要防止漂到海里去，他想要驾驶底盘低的车，请你告诉他，在不挂蹭到车底盘的前提下，他最低能驾驶底盘多低的车通过这段路。

由于车是 smzzl 自己改装的，因此 smzzl 将几乎所有的重量都分配在了车的接地处以防翻车。



Fig. 3. 【高手在民间单边桥极限会车】 <https://www.bilibili.com/video/BV1AV4y1q7KP>

在本题中，道路被抽象成如下图所示：左侧蓝色粗线的是海洋，中间黑色的是路面，右侧蓝色细线是墙壁（竖的是墙壁侧面，横着的是墙壁顶面）；红色的是车轮，绿色的是底盘，紫色的是车身。车辆允许两个轮胎都在路面上，或者一个轮胎在路面上另一个在墙壁的顶面上。车辆需要满足除了轮胎最下面的接地处，其他地方均不能和墙壁有交，恰好碰到也认为是有交（显然车不能穿模，这可是真实车辆!）。

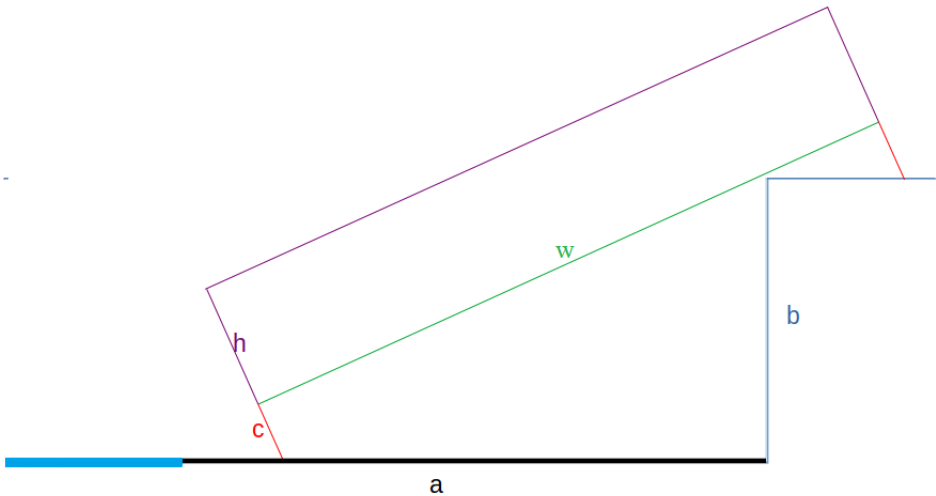


Fig. 4. 模型示意图

在本题中，车轮的大小也就代表的底盘高度，也就是图中的参数 c 。

给定车和道路的参数，请你告诉 smzzl 最低允许驾驶的车辆底盘高度。路面和车辆参数具体见题目输入部分。

Input

本题每一个测试点包含多组测试数据。

第一行一个整数 $T(1 \leq T \leq 1000)$ ，表示数据组数。

对于每一组数据，一行，4 个整数 $a, b, h, w(1 \leq a, b, h, w \leq 10^6)$ ，分别表示路面宽度，墙壁高度，车辆高度（不算底盘）和车宽。

Output

T 行，每行一个数表示该组数据的答案。如果无论如何无法通过则输出 “-1”（不含引号）。

在最低允许驾驶的车辆底盘高度存在的情况下，你的答案可以有至多 10^{-6} 的误差。形式化的说，假设你的答案是 a ，标准答案是 b ，则你的答案会被认为是正确的当且仅当 $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ 。

Example

standard input	standard output
8	0.4921567416
4 1 2 8	0.0000000000
7 5 3 3	0.0000000000
7 5 3 7	1.6286300478
7 5 3 12	-1
1 6 3 5	-1
1 6 3 6	2.2333296647
1 6 3 7	-1
5 8 5 5	

Problem K. str 进制

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

str，一个充满着神秘气息的字符串。

它既可以是某位 jiker 的姓名缩写，也可以表示字符串这个单词的缩写，由于其多变的特性，我们可以定义 str 进制字符串，下面的符合下面进位规则的进制称为该字符串下的 str 进制：

给出一个 str 进制字符串，每位上的数字 x 表示该位上是逢 x 进 1。

给定某个字符串 s ，再给出一个十进制下的整数 d ，求 d 在 str 进制下的表示值。

Input

第一行输入两个整数 m, n ($1 \leq m \leq 1000, 1 \leq n \leq 10$)，代表 str 进制字符串的长度和 d 的位数。

第二行输入一个字符串 s ($2 \leq s_i \leq 9$)，表示 str 字符串。

第三行输入一个整数，表示 d ($0 \leq d \leq 10^{10}$)。

数据保证 d 能够在 str 进制下被表示出来。

Output

输出一行一个 m 位的整数，为整数 d 在 str 进制下的表示。(包括前导零)

Example

standard input	standard output
4 1 2664 8	0020

Problem L. Equation

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

Given an integer M , find all integer values of x that satisfy the following conditions:

$$\begin{cases} x^2 \equiv x \pmod{M} \\ 0 \leq x < M \end{cases}$$

where $A \equiv B \pmod{M}$ denotes that expression A is congruent to expression B modulo M , meaning that the remainders of both sides when divided by M are equal.

Input

Each test contains multiple test cases. The first line contains the number of test cases T ($1 \leq T \leq 1000$). For each test case:

The only line contains only one integer M ($1 \leq M \leq 10^9$).

Output

For each test case, output two lines:

The first line outputs an integer n , representing the number of solutions to the equation.

The second line outputs n integers, listing the solutions in **ascending order** separated by spaces.

Example

standard input
4 6 114514 20231126 1
standard output
4 0 1 3 4 8 0 1 22165 35093 57257 57258 79422 92350 4 0 1 10115563 10115564 1 0

Note

In the first test case of example:

$$0^2 = 0 + 0 \times 6,$$

$$1^2 = 1 + 0 \times 6,$$

$$2^2 = 4 + 0 \times 6,$$

$$3^2 = 3 + 1 \times 6,$$

$$4^2 = 4 + 2 \times 6,$$

$$5^2 = 1 + 4 \times 6,$$

So, there are four solutions in total, which are 0, 1, 3 and 4.

Problem M. 小 H 的糖果

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

小 H 很喜欢吃糖。

有一天，小 L 给小 H 准备了一大串糖果，它们从前往后排成一行，每个糖果上都写有一个小写字母。由于单纯的吃糖显得太单调，于是小 L 制定了一个规则，小 H 只能选择一个糖果，然后依次吃掉它和它后面的所有糖果。吃掉的这串糖果的组成的字符串字典序越大，小 H 的满意度就越高。然而小 H 可不会这么轻易地就同意小 L 的规则，于是据理力争来了一个条件：可以任意选择恰好一个位置，将上面的糖果替换为写有任意一个小写字母的糖果（允许替换前后的糖果上的字母相同）。

现在，聪明的你告诉小 H，该如何替换糖果以及选择起始位置的糖果，得以让小 H 的满意度最高呢？

对于两个字符串 s_1 和 s_2 ，它们的长度分别为 n_1 和 n_2 。定义 s_1 的字典序比 s_2 的小，当且仅当其满足以下两个条件之一：

- 1. 如果存在一个位置 i ，使得对于所有 $j < i$ ， $s_{1,j} = s_{2,j}$ ，而 $s_{1,i} < s_{2,i}$ 。即在第一个不相等的位置上， s_1 的字母在字母表中的顺序靠前。
- 2. 如果 s_1 是 s_2 的前缀，且 $n_1 < n_2$ 。

Input

第一行输入一个 n ($1 \leq n \leq 5000$)，代表糖果的个数。
第二行输入一个字符串 s ($|s| = n$)，代表糖果上字母的序列。
题目保证 s 只包含小写字母。

Output

输出共一行一个字符串，表示让小 H 满意度最大的糖果串。

Examples

standard input	standard output
10 zzazzzabcd	zzzzzzabcd
8 azzzabcd	zzzzbcd