

2026年ICPC国际大学生程序设计竞赛 全国邀请赛（南昌） 暨江西省大学生程序设计竞赛

正式赛

2026 年 5 月 17 日



试题列表

A	马拉松的困难
B	UNO
C	旅行者的愿望
D	吃樱桃
E	完美无缺
F	一笔画
G	世界的意义
H	粑粑博弈
I	搞快点搞快点
J	GCD 之旅
K	祖传代码
L	小 Z 的函数
M	构造王坝

本试题册共 13 题，21 页。
如果您的试题册缺少页面，请立即通知志愿者。

Problem A. 马拉松的困难

本题题目背景与题意无关，如果不想看请直接跳到最后一段。

公元前 490 年希腊马拉松战役取胜后，传令兵菲迪皮茨从马拉松平原奔跑约 40 公里回雅典报捷，宣告胜利后不幸离世；1896 年首届雅典现代奥运会为纪念此事设立马拉松项目，初始距离约 40 公里，1908 年伦敦奥运会调整路线定为 42.195 公里，1921 年被国际田联确立为通用标准，女子马拉松则在 1984 年洛杉矶奥运会成为正式竞赛项目。

马拉松极易造成体能极限透支，在 30 至 35 公里易出现糖原耗尽的撞墙期，引发身体与意志双重崩溃；长时间奔跑易给膝盖、脚踝、小腿等部位带来伤病风险，同时开局配速难把控、后半程充满孤独的意志煎熬，再加天气变化带来的体感压力、补给时机不当易引发肠胃不适和电解质紊乱，且参赛对日常系统备赛耐力与力量训练门槛要求极高，缺乏准备强行参赛极易受伤。

然而，马拉松最大的难点还来自于极低的中签率。热门马拉松的中签率往往都低于 10%，新手甚至低至 2%。

如果你是马拉松新手，你认为跑马拉松最大的困难是什么呢！

对于每个中签因子 N 和报名序号 M ，若 N 能整除 M 即表示中签。告诉你报名序号和中签因子，试问你能中签吗？

Input

第一行一个正整数 T ($1 \leq T \leq 100$)，代表测试实例的个数。

接下来有 T 行，每行代表一组测试实例，为两个正整数 M, N ($1 \leq M, N \leq 100000$)，表示你报名的序号 M 和中签因子 N 。

Output

每组测试实例输出一行，为一个整数，中签输出 1，没中签输出 0。

Example

standard input	standard output
2	1
2026 2026	0
2 2025	

Problem B. UNO

Troubadour_Ggmz 和 Fengmi123 正在玩一局特殊的 UNO 游戏。UNO 牌共有四种颜色：红色、黄色、绿色、蓝色，分别用 R, Y, G, B 表示。每种颜色包含 n 张牌，点数是 1 到 n 的任意整数；可能有两张牌具有相同的颜色和点数。整副牌共有 $4n$ 张。

游戏开始时，Troubadour_Ggmz 将全部 $4n$ 张牌排成一行，然后 Fengmi123 可以执行任意多次以下操作：选择相邻的两张牌，如果它们颜色相同或者点数相同，则交换这两张牌的位置。

游戏的目的是将所有牌整理成颜色和点数同时有序的状态，即：

- 颜色顺序为 $R < Y < G < B$ ，即所有红色牌出现在所有黄色牌之前，所有黄色牌出现在所有绿色牌之前，所有绿色牌出现在所有蓝色牌之前；
- 对于同种颜色的牌，点数序列必须是非递减的。

Fengmi123 想要知道，她是否可以通过若干次（包括零次）允许的操作，达成目标状态？

Input

有多组测试数据。第一行输入一个整数 t ($1 \leq t \leq 1000$)，表示测试数据组数。对于每组测试数据：

第一行一个正整数 n ($1 \leq n \leq 1000$)。

接下来 $4n$ 行，每行包含一个正整数 x_i 和一个字符 c_i ，其中 $1 \leq x_i \leq n$ ， $c_i \in \{R, Y, G, B\}$ ，依次表示第 i 张牌的点数和颜色。输入保证每种颜色恰好出现 n 次，但同色同点数的牌可能出现多次。

保证所有测试数据的 n 之和不超过 1000。

Output

对于每组测试数据，如果可以经过若干次操作达到目标状态，输出“YES”，否则输出“NO”。

你可以输出任意大小写形式的答案。例如，字符串“yEs”、“yes”、“Yes”、“YES”等都会被接受为肯定回答。

Example

standard input	standard output
2	YES
2	NO
1 R	
2 Y	
2 R	
1 G	
1 Y	
2 B	
2 G	
1 B	
2	
1 R	
2 Y	
1 Y	
2 B	
2 R	
1 G	
2 G	
1 B	

Problem C. 旅行者的愿望

构造王国是著名的旅游胜地，每年都会吸引数以万计的游客前来旅游。王国由 n 座城镇和 m 条单向道路组成，每条道路长度均为 1。保证初始的 m 条道路中没有重复的道路，也没有从一座城镇通向自身的道路。

后来由于旅游业的持续繁荣，构造王国决定扩建道路。从第 1 天开始，在接下来的 k 天里，每天都会新建一条从 x 到 y 的单向道路。由于规划不当，可能出现修建一条已经存在的道路，或者修建一条自己到自己的道路的情况。

枫蜜是一位任性的旅行者，她有 q 个旅行愿望和一个心愿距离 w 。每个愿望有一个起点 s 和终点 t (s_i 和 t_i 可能相同)。她想知道，对于每个愿望，最早在第几天的时候，可以从城镇 s 出发，在允许多次经过城镇和道路的前提下，恰好走 w 的距离到达城镇 t ，或报告无法实现。请你帮她解决这个问题。

Input

第一行包含两个整数 n 和 m ($2 \leq n \leq 200, 1 \leq m \leq n(n-1)$)，表示城镇数量和初始道路数量。

接下来 m 行，每行包含两个整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$)，表示一条从 u_i 到 v_i 的有向道路。

接下来一行包含一个整数 k ($1 \leq k \leq n^2$)，表示接下来要修建的道路数量。

接下来 k 行，每行包含两个整数 x_i 和 y_i ($1 \leq x_i, y_i \leq n$)，表示第 i 天新建一条从 x_i 到 y_i 的有向道路。

接下来一行包含两个整数 q 和 w ($1 \leq q \leq n^2, 1 \leq w \leq 10^9$)，分别表示愿望数量和心愿距离。

接下来 q 行，每行包含两个整数 s_i 和 t_i ($1 \leq s_i, t_i \leq n$)，描述第 i 个愿望的起点和终点。

Output

对于每个愿望，输出一个整数，表示最早能够实现该愿望的天数（即在那一天修建完道路后，恰好存在一条长度为 w 的路径从 s 走到 t ）。如果永远无法实现，输出 -1 。

Example

standard input	standard output
6 3	-1
1 3	2
2 3	0
3 1	2
4	4
4 5	
5 5	
2 5	
5 6	
5 8	
5 1	
5 5	
1 1	
4 5	
4 6	

Note

在样例中，心愿距离 $w = 8$ 。

- 对于第一个愿望，可以证明无法实现。
- 对于第二个愿望，在修建了 $4 \rightarrow 5$ 和 $5 \rightarrow 5$ 这两条道路后，可以沿着城镇序列 $[5, 5, 5, 5, 5, 5, 5, 5]$ 行走，从而到达终点。
- 对于第三个愿望，初始时即可沿着城镇序列 $[1, 3, 1, 3, 1, 3, 1, 3]$ 走到终点。
- 对于第四个愿望，在修建了 $4 \rightarrow 5$ 和 $5 \rightarrow 5$ 这两条道路后，可以沿着城镇序列 $[4, 5, 5, 5, 5, 5, 5, 5]$ 走到终点。
- 对于第五个愿望，在修建了 $4 \rightarrow 5, 5 \rightarrow 5, 2 \rightarrow 5, 5 \rightarrow 6$ 这些道路后，可以沿着城镇序列 $[4, 5, 5, 5, 5, 5, 5, 6]$ 走到终点。

Problem D. 吃樱桃

江桥喜欢吃樱桃！在樱桃还未成熟的时候，他就早早来到了樱桃树前...

一棵樱桃树由 n 个结点（编号 1 到 n ）和 $n - 1$ 条树枝组成，通过这些树枝把所有结点都连接了起来。

江桥根据每颗樱桃的大小、色泽、甜度等指标把整棵樱桃树上的樱桃分为了 m 种（编号 1 到 m ），其中第 i 个结点上结了 k_i 种樱桃。

定义一条**主干枝**为樱桃树的根结点到某个边缘结点的不经过重复结点的路径。

即对于根节点 u ， u 到某个边缘结点 v 的主干枝为一个结点序列 $[u = x_0, x_1, x_2, \dots, x_s = v]$ ，其中 x_i 号结点与 x_{i+1} ($i < s$) 号结点通过一条树枝连接，且 $i \neq j$ 时 $x_i \neq x_j$ 。

边缘结点指与其他结点仅有一条树枝连接的结点。特殊地，**根结点不是边缘结点**。

显然对于每个边缘结点，其到根节点的主干枝是唯一确定的。

江桥可以施展魔法。**每次魔法**可以选择一个边缘结点并**任意改变**该结点上樱桃的种类和种类数量（包括将其变为无樱桃）。

他想知道，对于 $1, 2, \dots, n$ 号结点分别作为根节点，**最少需要施展多少次魔法**，才能让每条主干枝上包含的**樱桃种类**均相同。

Input

第一行包含两个整数 n 和 m ($2 \leq n \leq 2 \times 10^5, 1 \leq m \leq 6$)，表示树中结点个数和樱桃种类数。

接下来 $n - 1$ 行，每行包含两个整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$)，表示树中的一条树枝。

接下来 n 行，第 i 行的第一个整数为非负整数 k_i ($0 \leq k_i \leq m$)，表示 i 号结点所结的樱桃种类数。如果 $k_i > 0$ ，则后面跟随 k_i 个互不相同的正整数 $t_{i,1}, t_{i,2}, \dots, t_{i,k_i}$ ($1 \leq t_{i,j} \leq m$)，表示樱桃的种类。

输入数据保证所有结点可以通过树枝相互到达。

Output

一行 n 个非负整数，第 i 个数表示 i 号结点为根时的答案。

Example

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

Problem E. 完美无缺

枫蜜正在挑战一款音乐游戏的高难关卡。关卡中有 n 个按顺序下落的音符，第 i 个音符的基础分值为 a_i 。当玩家击中前 i 个音符后，**累计得分** $s_i = \sum_{k=1}^i a_k$ 将决定连击特效的触发条件。

具体来说，第 i 个音符有一个得分目标区间 $[l_i, r_i]$ ，只有 s_i 落在该区间内，才能触发完美的连击特效演出，否则特效链会中断。

作为“区域增幅”道具的拥有者，枫蜜可以任意次使用该道具：每次使用可以选择一段连续的音符 $[l, r]$ ，并将这些音符的基础分值全部增加 1。道具的使用次数会影响最终的关卡评分，因此枫蜜希望用最少的次数达成全程完美连击。

请你帮她计算，最少需要使用多少次区域增幅道具，才能让每个音符 i 的累计得分 s_i 都满足 $l_i \leq s_i \leq r_i$ ，或报告无解。

Input

有多组测试数据。第一行输入一个整数 t ($1 \leq t \leq 5000$)，表示测试数据组数。对于每组测试数据：

第一行包含一个整数 n ($1 \leq n \leq 5000$)，表示音符的数量。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$)，表示每个音符的基础分值。

接下来 n 行，每行包含两个整数 l_i 和 r_i ($0 \leq l_i \leq r_i \leq 10^{18}$)，表示在第 i 个音符击中后的累计得分目标区间。

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，输出一个整数，表示最少道具使用次数。若无法达成目标，输出 -1 。

Example

standard input	standard output
2	4
3	-1
6 2 8	
8 16	
16 32	
28 36	
2	
0 0	
1 1	
0 0	

Note

对于样例的第一组数据，一种可行的操作方案如下：对区间 $[1, 3]$ 使用 4 次道具，原基础分值序列变为 $[10, 6, 12]$ ，累计得分为 $[10, 16, 28]$ ，满足条件。总操作次数为 4。可以证明不存在操作次数更少的方案。

对于样例的第二组数据，可以证明不存在可行的操作方案。

Problem F. 一笔画

枫蜜最近迷上了一款名为“一笔画”的小游戏。在这个游戏中，地图是一张包含 n 个节点和 m 条边的无向图，图中可能存在重边和自环，且不一定连通。玩家需要从某个节点出发，沿着图中的边行走，恰好经过每条边一次，最后可以停留在任意节点。当所有边都被经过恰好一次后，游戏立刻结束。

这款游戏增加了一个特殊的“传送”机制：有 k 对互不相同的“跳跃点”，每对 (a, b) 具有如下性质：每当玩家经过一条边到达节点 a 或节点 b 时，会立刻被传送到该点对中的另一个节点，并在传送后继续行动。

需要特别注意的是：

- 起点不会触发传送（即如果起点是一个跳跃点，从该点出发时不会发生传送）；
- 终点也不会触发传送（即如果最后一步到达一个跳跃点，游戏立即结束，不再进行传送）。

枫蜜想知道，是否存在一条路径，使得玩家从某个节点出发，经过每条边恰好一次，并严格遵守上述跳跃规则。如果存在，请输出任意一条这样的路径（以经过的节点序列表示），或报告无解。

Input

第一行包含一个正整数 t ($1 \leq t \leq 5 \times 10^4$)，表示测试数据组数。

对于每组数据：

第一行包含三个正整数 n 、 m 和 k ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^6$, $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$)，分别表示节点的数量、边的数量和跳跃点对的数量。

接下来 m 行，每行包含两个正整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n$)，描述一条无向边（可能存在重边和自环）。

接下来 k 行，每行包含两个正整数 a_i 和 b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$)，表示 (a_i, b_i) 是一对跳跃点。保证所有 a, b 互不相同（即每个节点至多出现在一对跳跃点中）。

保证所有数据的 n 之和不超过 10^5 ， m 之和不超过 10^6 。

Output

对于每组数据，如果存在满足条件的路径：

第一行输出一个整数 L ，表示路径经过的节点个数（包括起点和终点）；

第二行输出 L 个整数，两两之间以一个空格分隔，表示依次经过的节点编号。

如果不存在，输出一行 -1 。

注意：路径序列中需要记录实际到达的节点（包括传送后到达的节点）。例如，从节点 1 走到 2（假设 2 是跳跃点，传送到 3），然后从 3 走到 4，则输出序列为 $[1, 2, 3, 4]$ 。

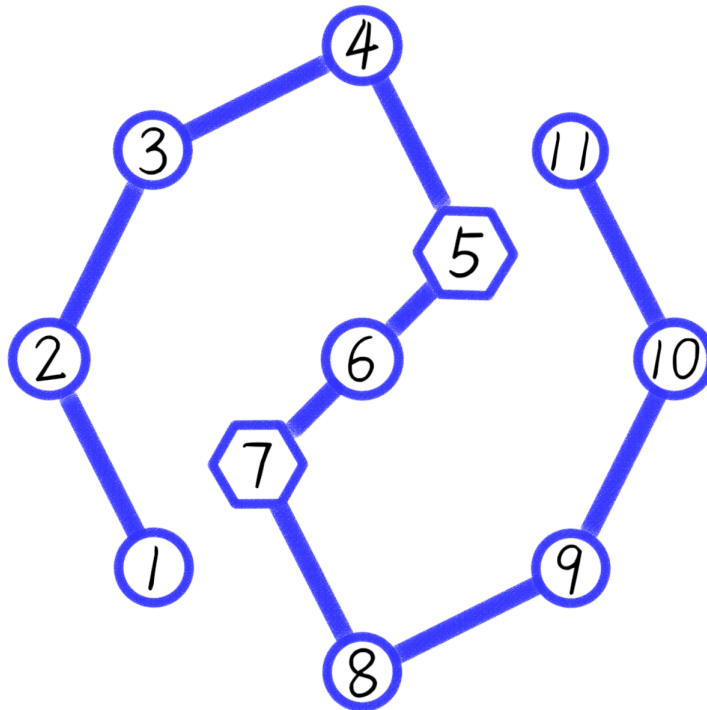
Example

standard input	standard output
2	13
11 10 1	1 2 3 4 5 7 6 5 7 8 9 10 11
1 2	30
2 3	5 6 7 8 9 3 4 5 4 1 9 3 2 9 3 2 1 9 3
3 4	1 8 9 3 1 7 6 1 5 1 7
4 5	
5 6	
6 7	
7 8	
8 9	
9 10	
10 11	
5 7	
9 24 1	
1 2	
1 3	
1 3	
1 4	
1 5	
1 5	
1 6	
1 7	
1 7	
1 8	
1 9	
1 9	
2 3	
2 3	
3 4	
4 5	
4 5	
5 6	
6 7	
6 7	
7 8	
8 9	
8 9	
9 2	
9 3	

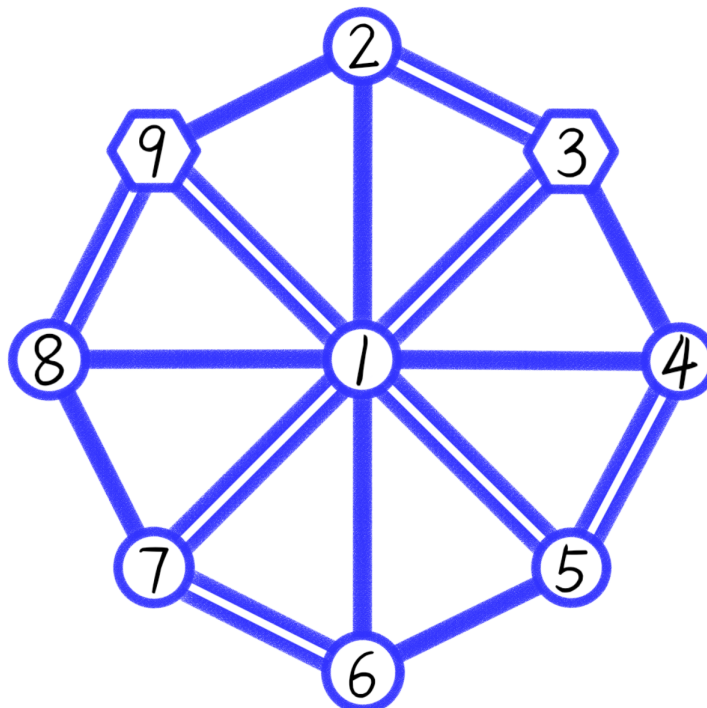
Note

对于样例的第一组数据, (5, 7) 是一对跳跃点, 如图所示。

一条可行的路径是: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ (传送) $\rightarrow 7 \rightarrow 6 \rightarrow 5$ (传送) $\rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11$ 。



对于样例的第二组数据, 如图所示:



Problem G. 世界的意义

世界的意义必定存在于世界之外

世界中的一切事物如其所存在般而存在，如其所发生般而发生

世界之中不存在价值

——《逻辑哲学论》

枫蜜给了你一个长度为 n 的字符串 s ，仅由 ‘(’ 和 ‘)’ 组成。

你需要支持 q 次操作：

1. 翻转第 k 个括号，即将 ‘(’ 变为 ‘)’, ‘)’ 变为 ‘(’;
2. 查询区间 $[l, r]$ 内最长的合法括号子串的长度。

形式化来说，你要求：

$$f(l, r) = \max(\{j - i + 1 \mid l \leq i \leq j \leq r \wedge \text{valid}(i, j)\} \cup \{0\})$$

其中 $\text{valid}(i, j)$ 表示子串 $s[i..j]$ 是一个合法的括号序列。

在本题中，认为以下的括号序列是合法的：

1. 空序列是一个合法序列。
2. 如果 A 是一个合法序列，则 (A) 也是一个合法序列。
3. 如果 A 和 B 都是合法序列，则 AB 也是一个合法序列。

Input

第一行包含两个正整数 n 和 q ($1 \leq n, q \leq 2 \times 10^5$)，表示给定括号串的长度和操作次数。

第二行包含一个长度为 n 的字符串 s ，仅由 ‘(’ 和 ‘)’ 组成。不保证给定的字符串是一个合法的括号序列。

接下来 q 行，每行为以下两种操作之一：

- 1 k ($1 \leq k \leq n$): 翻转第 k 个括号。
- 2 l r ($1 \leq l \leq r \leq n$): 查询区间 $[l, r]$ 内最长的合法括号子串的长度。

Output

对于每个操作 2，输出一个整数，表示答案。

Examples

standard input	standard output
8 5 () (()) 2 1 8 1 4 2 1 8 1 7 2 1 8	8 4 4
12 6 ((()) (())) 2 1 12 1 3 2 1 12 1 8 2 1 12 2 4 9	12 10 12 2

Note

样例 1:

- 初始字符串: `() ( ( ) )`;
- 第一次查询区间 `[1, 8]`, 最长合法括号子串是 `() ( ( ) )` (位置 `1 - 8`), 长度为 `8`;
- 翻转第 4 个括号: `() ( ) ( )`;
- 第二次查询区间 `[1, 8]`, 最长合法括号子串是 `() ( )` (位置 `1 - 4`), 长度为 `4`;
- 翻转第 7 个括号: `() ( ) ( ( ) )`;
- 第三次查询区间 `[1, 8]`, 最长合法括号子串是 `() ( )` (位置 `1 - 4`), 长度为 `4`。

样例 2:

- 初始字符串: `(( ( ) ) ( ( ) ) )`;
- 第一次查询区间 `[1, 12]`, 最长合法括号子串是整个字符串, 长度为 `12`;
- 翻转第 3 个括号: `(( ( ( ) ) ( ( ) ) )`;
- 第二次查询区间 `[1, 12]`, 最长合法括号子串是 `(( ( ) ) ( ( ) ) )` (位置 `3 - 12`), 长度为 `10`;
- 翻转第 8 个括号: `(( ( ( ) ) ( ( ) ) ( ) )`;
- 第三次查询区间 `[1, 12]`, 最长合法括号子串是 `(( ( ( ) ) ( ( ) ) ( ) )` (位置 `1 - 12`), 长度为 `12`;
- 第四次查询区间 `[4, 9]`, 最长合法括号子串是 `( )` (位置 `4 - 5`), 长度为 `2`。

Problem H. 粑粑博弈

Alice 和 Bob 两个人轮流进行操作，Alice 先手，在第一次操作前，已经给定一个长度为 n 的非负整数数组 $a_1, a_2, \dots, a_n$ ，和一个正整数 k ，每次操作，操作人需要选择两个整数 x, y 满足 $0 \leq x \leq k$ 且 $1 \leq y \leq n$ 且 $x \leq a_y$ ，并把 k 修改为 x ， a_y 修改为 $a_y - x$ ，若某玩家选择 $x = 0$ ，则该玩家立即输掉游戏。问谁有必胜策略？

Input

本题有多组测试数据，第一行输入 t ($1 \leq t \leq 10^5$)，代表测试数据组数。

对于每组测试数据：

第一行输入两个正整数 n 和 k ($1 \leq n \leq 10^6, 1 \leq k \leq 10^9$)。

第二行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$)。

保证单次输入数据里面所有组的 n 的和小于等于 2×10^6 。

Output

对于每组数据，输出一行一个字符串，如果对于这组数据输入所描述的游戏，先手（又称 Alice）有必胜策略，则输出 “Alice”，否则输出 “Bob”。

Example

standard input	standard output
3	Alice
2 2	Alice
1 2	Bob
4 5	
15 15 8 9	
11 2	
4 2 9 12 0 15 9 14 5 6 12	

Note

对于样例第一组数据，Alice 第一轮能保证必胜的选择有且仅有 $(x, y) = (1, 1)$ 和 $(x, y) = (1, 2)$ 。

Problem I. 搞快点搞快点

你玩过蛋仔派对吗？inf 很喜欢玩蛋仔派对。

四个蛋仔叠成一个四层高的蛋仔柱，初始底部位于坐标 0。从下到上依次记为 A、B、C、D，其中 D 是最顶端的蛋仔。

赛道总长度为 l ，包含 l 段长度为 1 的线段，线段 i 连接坐标 $i - 1$ 和 i ($1 \leq i \leq l$)。

第 i 段线段的地形由字符 S_i 给出。若蛋仔柱从坐标 $i - 1$ 走到坐标 i ，需要花费：

- 平地 0: t_0 秒；
- 泥潭 1: t_1 秒；
- 加速带 2: t_2 秒。

蛋仔柱可以进行两种操作：

- **普通前进：** 若当前蛋仔柱位于坐标 $x < l$ ，则可整体前进至坐标 $x + 1$ 。这次前进的耗时只与第 $x + 1$ 段地形有关。
- **超级起步：** 若当前坐标 x 处有一个包含 j 个蛋仔的蛋仔柱 ($j \geq 2$)，则最底部的蛋仔可以把上面的 $j - 1$ 个蛋仔瞬间扔到坐标 $\min(l, x + d)$ ，其中 d 为一个给定的正整数。这个操作不触发地形耗时，即耗时为 0。操作后，原地最底部的蛋仔被淘汰，原位置变为空；坐标 $\min(l, x + d)$ 处形成一个包含 $j - 1$ 个蛋仔的新蛋仔柱，剩余蛋仔的上下顺序保持不变。

请计算让初始时最顶端的蛋仔 D 到达终点坐标 l 所需要的最短时间。

Input

第一行包含两个正整数 l 和 d ($1 \leq d < l \leq 2 \times 10^5$)。

第二行包含三个正整数 t_0, t_1, t_2 ($1 \leq t_2 < t_0 < t_1 \leq 10^9$)。

第三行包含一个长度为 l 的字符串 S ，仅由字符 0、1、2 组成。

Output

输出一个整数，表示最短总时间。

Examples

standard input	standard output
10 2 10 20 5 0000000000	40
12 3 10 100 1 111000222000	3
10 2 10 100 1 2220110000	13
8 3 10 100 1 00000000	0

Note

对于样例一：一种最优方案是连续发动 3 次超级起步到达坐标 6，再走完最后 4 段平地，总耗时为 $4 \times 10 = 40$ 。

对于样例二：一种最优方案是先连续两次超级起步到达坐标 6，步行走过 3 段加速带到达坐标 9，最后再超级起步到终点，总耗时为 $1 + 1 + 1 = 3$ 。

对于样例三：一种最优方案是先走过前 3 段加速带和 1 段平地到达坐标 4，再连续发动 3 次超级起步到达终点，总耗时为 $1 + 1 + 1 + 10 = 13$ 。

对于样例四：连续发动 3 次超级起步，坐标变化为 $0 \rightarrow 3 \rightarrow 6 \rightarrow 8$ （最后一次触发 $\min(8, 6 + 3)$ 限制），全程不需要步行，总耗时为 0。

Problem J. GCD 之旅

在神奇的 Numerian 魔法王国中，有 n 座城市，每座城市都拥有自己的魔法能量值 a_i 。国王在每两座城市之间都修建了双向道路，其中连接城市 u 和城市 v 的道路的守护符强度（Guardian Charm Degree，简称 GCD）等于 $\gcd(a_u, a_v)$ ，即两座城市魔法能量值的最大公约数。

旅行者从城市 x 前往城市 y 时，其旅程的魔法免疫指数（Magical Immunity Number，简称 MIN）等于所经过的所有道路中 GCD 的最小值。王国的皇家顾问认为一次旅程是“强免疫”的，当且仅当它的 MIN 达到了可能的最大值。

你的任务是回答国王的 q 次询问。对于每次询问 (x, y) ，输出从城市 x 到城市 y 的所有路径中，可能的最大 MIN。

特别地，如果 $x = y$ ，旅行者没有离开城市，此时 MIN 等于该城市自身的魔法能量值 a_x 。

Input

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

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$)，表示每座城市的魔法能量值。

接下来 q 行，每行包含两个整数 x 和 y ($1 \leq x, y \leq n$)，表示一次询问。

Output

对于每次询问，输出一行一个整数，表示可能的最大 MIN。

Example

standard input	standard output
6 3	7
6 15 22 33 35 63	3
5 6	15
3 5	
2 2	

Note

对于样例，魔法能量值 $a = [6, 15, 22, 33, 35, 63]$ 。

第 1 组询问：最优路径为 $5 \rightarrow 6$ ，答案为 $\gcd(35, 63) = 7$ 。

第 2 组询问：最优路径为 $3 \rightarrow 4 \rightarrow 6 \rightarrow 5$ ，三条边的 GCD 分别为 $\gcd(22, 33) = 11$ ， $\gcd(33, 63) = 3$ 和 $\gcd(63, 35) = 7$ ，答案为 $\min(11, 3, 7) = 3$ 。

第 3 组询问：由于 $x = y$ ，旅行者没有离开城市，答案为 $a_2 = 15$ 。

Problem K. 祖传代码

你有一段祖传代码：

```
for  $t = 1$  to  $k$  do
     $a_{x_1} \leftarrow (a_{u_1} \times a_{v_1}) \bmod MOD$ 
     $a_{x_2} \leftarrow (a_{u_2} \times a_{v_2}) \bmod MOD$ 
     $a_{x_3} \leftarrow (a_{u_3} \times a_{v_3}) \bmod MOD$ 
     $\vdots$ 
     $a_{x_m} \leftarrow (a_{u_m} \times a_{v_m}) \bmod MOD$ 
end for
```

其中 $MOD = 998\,244\,353$ ，且 $x_i, u_i, v_i \in \{1, \dots, n\}$ ， $i = 1, \dots, m$ 。

给定变量 $a_1, a_2, \dots, a_n$ 的初始值，求这个循环运行 k 次后这些变量的最终值。你需要对 q 组这样的询问进行回答。

Input

第一行包含三个整数 n, m 和 q ($1 \leq n \leq 200, 1 \leq m \leq 2 \times 10^5, 1 \leq q \leq 200$)。

接下来 m 行，每行包含三个整数 x_i, u_i, v_i ($1 \leq x_i, u_i, v_i \leq n$)。

接下来 q 行，每行表示一组询问。每组询问包含 $n + 1$ 个整数：

- 循环次数 k ($1 \leq k \leq 10^6$),
- n 个变量的初始值 $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 998\,244\,353$)。

Output

对于每组询问，输出一行 n 个整数，即代码执行后 $a_1, a_2, \dots, a_n$ 的最终值，对 $998\,244\,353$ 取模。

Example

standard input
7 8 4 3 1 2 4 3 3 5 6 7 7 1 3 2 2 2 6 6 2 1 5 6 7 7 7 1 2 1 1 1 1 1 1 2 1 2 3 4 5 6 7 3 1 1 0 1 0 1 0 100 11 45 14 19 19 8 10
standard output
1 1 2 4 1 1 16 36864 16 4032 16257024 96 384 227524445 0 1 1 1 0 1 1 244858606 337827833 584144609 343808937 412777304 279017446 926699892

Problem L. 小 Z 的函数

小 Z 在高中曾经沉迷于欧拉函数 φ *等数论函数。

某一天，小 Z 打算和他的机器人朋友玩这样一个游戏：首先小 Z 会给出一个长度为 n 的排列[†] p 。随后小 Z 会多次向朋友提问，对于区间 $[l, r]$ ($1 \leq l < r \leq n$)，最大的 $\varphi(p_i \times p_j)$ ($l \leq i < j \leq r$) 的值是多少？

形式化地，对于每一组询问 l, r ，你都需要计算：

$$\max_{l \leq i < j \leq r} \varphi(p_i \times p_j)$$

小 Z 已经解决了 0 个询问，现在请你帮帮他，完成剩下的 q 个询问的计算！

Input

第一行包含一个正整数 n ($2 \leq n \leq 5 \times 10^5$)，表示排列的长度。

第二行包含 n 个互不相同的正整数 $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$)，表示给定的排列 p 。

第三行包含一个正整数 q ($1 \leq q \leq 5 \times 10^5$)，表示询问的个数。

接下来 q 行，每行包含两个正整数 l_i 和 r_i ($1 \leq l_i < r_i \leq n$)，表示第 i 组询问的左右端点。

Output

对于每一组询问，输出一个整数，表示对应的最大值。

Example

standard input	standard output
6	2
3 2 5 4 1 6	8
8	8
4 5	4
2 6	8
3 5	8
2 3	2
3 4	8
1 5	
4 5	
2 6	

* $\varphi(n)$ 表示不超过 n 的正整数中与 n 互质的数的个数。

[†]一个长度为 n 的排列是一个包含 n 个整数的序列，其中 1 到 n 的每个整数恰好出现一次。例如， $[2, 3, 1, 5, 4]$ 是一个排列，而 $[1, 2, 2]$ 不是（2 出现了两次）， $[1, 3, 4]$ 也不是（ $n = 3$ 但出现了 4）。

Problem M. 构造王坝

你正在管理构造王坝（Kingdam of Construction）的阀门矩阵。

给定一个 $n \times m$ （ n 和 m 均为偶数）的矩阵 a ， a 里的元素均为 1 到 $n \times m$ 之间的正整数且两两不同。

你可以进行以下两种操作（任意顺序）：

- 选择第 i ($1 \leq i \leq n$) 行，将其左右翻转，即对于所有 $1 \leq j \leq \frac{m}{2}$ ，交换 $a_{i,j}$ 与 $a_{i,m-j+1}$ ；
- 选择第 j ($1 \leq j \leq m$) 列，将其上下翻转，即对于所有 $1 \leq i \leq \frac{n}{2}$ ，交换 $a_{i,j}$ 与 $a_{n-i+1,j}$ 。

求能否经过有限次操作把 a 变为 $n \times m$ 的目标矩阵 b ，其中 b 按行优先顺序依次填入 $1, 2, \dots, n \times m$ ，即 $b_{i,j} = (i-1) \times m + j$ 。如果可以，请构造一个操作次数不超过 9961 的操作序列（也可以不进行任何操作，即操作序列为空）。可以证明，如果存在一个操作序列，则必存在一个操作次数不超过 9961 的操作序列。

Input

有多组测试数据。输入的第一行包含一个正整数 t ($1 \leq t \leq 25$)，表示测试数据的组数。对于每组测试数据：

第一行包含两个正整数 n 和 m ($2 \leq n \leq 134, 2 \leq m \leq 134$)，分别表示矩阵 a 的行数与列数。

接下来 n 行，每行包含 m 个正整数 $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ，表示矩阵 a 的元素。

保证 n 和 m 均为偶数；保证 1 到 $n \times m$ 之间的每个正整数在矩阵 a 中恰好出现一次。

Output

对于每组测试数据：

如果可以在有限次操作内将矩阵 a 变为目标矩阵 b ，在第一行中打印 “Yes”。

接下来一行包含一个非负整数 cnt ，表示你的操作次数。你需要保证 $0 \leq cnt \leq 9961$ （对于每组测试数据，即对所有测试数据的 cnt 之和没有限制）。

接下来 cnt 行，每行包含两个整数 op 和 idx ($op \in \{0,1\}$ ，若 $op = 0$ 则 $1 \leq idx \leq n$ ，否则 $1 \leq idx \leq m$)， $op = 0$ 表示将第 idx 行翻转， $op = 1$ 表示将第 idx 列翻转。

如果不存在合法的操作方案，仅输出一行包含 “No”。

你可以输出任意大小写形式的答案。例如，字符串 “yEs”、“yes”、“Yes”、“YES” 均会被接受为肯定回答。

Example

standard input	standard output
4	Yes
2 4	2
4 3 2 1	0 1
8 7 6 5	0 2
2 4	Yes
1 3 2 4	5
5 6 7 8	1 2
4 2	0 1
1 2	1 2
3 4	0 1
7 8	1 2
5 6	No
4 4	No
4 15 14 16	
8 6 11 12	
5 7 10 9	
13 3 2 1	

Note

对于样例的第二组数据，操作过程如下：

$$\begin{array}{ccccc} & & \xrightarrow{\text{column, 2}} & & \xrightarrow{\text{row, 1}} \\ \begin{array}{c} \left[\begin{array}{cccc} 1 & 3 & 2 & 4 \\ 5 & 6 & 7 & 8 \end{array} \right] \\ \xrightarrow{\text{column, 2}} \left[\begin{array}{cccc} 4 & \mathbf{3} & 6 & 1 \\ 5 & \mathbf{2} & 7 & 8 \end{array} \right] \end{array} & \xrightarrow{\text{row, 1}} & \begin{array}{c} \left[\begin{array}{cccc} 1 & \mathbf{6} & 2 & 4 \\ 5 & \mathbf{3} & 7 & 8 \end{array} \right] \\ \left[\begin{array}{cccc} \mathbf{1} & \mathbf{6} & \mathbf{3} & \mathbf{4} \\ 5 & 2 & 7 & 8 \end{array} \right] \end{array} & \xrightarrow{\text{column, 2}} & \begin{array}{c} \left[\begin{array}{cccc} \mathbf{4} & \mathbf{2} & \mathbf{6} & \mathbf{1} \\ 5 & 3 & 7 & 8 \end{array} \right] \\ \left[\begin{array}{cccc} 1 & \mathbf{2} & 3 & 4 \\ 5 & \mathbf{6} & 7 & 8 \end{array} \right] \end{array} \end{array}.$$

对于样例的第三与第四组数据，目标矩阵 b 为 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$ 和 $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$ ，可以证明这两种情况均无解。