

北京理工大学第十五届“连山科技”程序设计大赛

The 15-th BIT Campus Programming Contest

Sponsored by



网络选拔赛

Online Round



题目列表

Problem List

A	“大”模拟
B	大采购
C	Changing Game
D	欧拉树
E	简单的计算几何
F	Ironhead 的推免资格 (传说)
G	题目册就是给你剪拼图的?
H	Prefix Sum
I	shenyunhan Loves Palindrome
J	屠龙勇者 ErvinXie
K	Strange Game
L	“我为什么要升级?”
M	这局我觉得你能赢

请勿在比赛开始前翻阅试题!

Do not open before the contest has started.

2020 年 11 月 14 日

参赛选手赛前须知

纪律：比赛开始前不要触碰任何东西！



1. 比赛细节相关

- (1) 每题的代码长度限制为 256 KB，每题仅限提交 1 个文件
- (2) Java jdk 版本为 11.0.7
- (3) 选手页面点击题目列表，可以查看时空限制、下载样例文件
- (4) 比赛中遇到题目相关的问题，请点击提问并发表问题
- (5) 比赛中遇到非题目相关的问题，请点击提问并发表问题或直接联系主办方
- (6) 建议严格按照题面要求格式输入输出，行末不要有多余的空格
- (7) C/C++代码中 main 函数必须定义为 int 型，且最终返回值为 0
- (8) 所有题目的输入文件换行符均为单'\n'

2. 编译参数

C gcc -x c -Wall -O2 -static -pipe -o "\$DEST" "\$@" -lm
C++ g++ -x c++ -Wall -O2 -std=c++14 -static -pipe -o "\$DEST" "\$@"
Java javac -encoding UTF-8 -sourcepath . -d . "\$@"

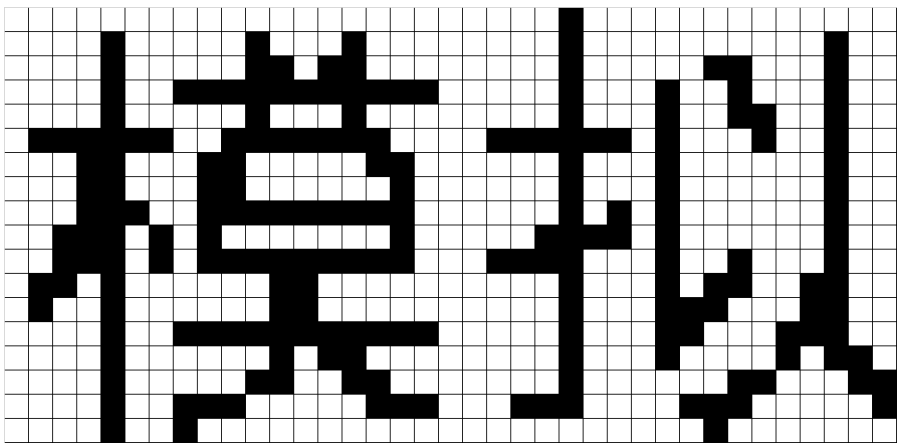
3. 评测结果返回说明

PENDING	提交正在评测或等待评测，请耐心等待。
CORRECT	答案正确！
WRONG-ANSWER	错误的结果，请检查自己的算法是否正确，边界条件是否考虑完全等。
RUN-ERROR	运行时错误，请检查自己是否存在数组越界、除 0、返回非 0 等情况等。 注：MEMORY-LIMIT 也有可能返回为 RUN-ERROR。
TIMELIMIT	超出时间限制，请检查自己是否存在死循环，算法可否优化等。
OUTPUT-LIMIT	输出过长，请检查自己的算法是否正确，是否死循环输出等。
MEMORY-LIMIT	超出空间限制，请检查自己的数组是否过大等。
COMPILER-ERROR	编译错误，请检查自己的代码的语法错误。
TOO-LATE	提交超时，比赛已经结束。

Problem A. “大” 模拟

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

LIT (Liangxiang University of Technology) 的学生组织智障算法团队要组织关于模拟算法的专题讲座，现场需要悬挂多张“模拟”二字的横幅，横幅具体的样式如下：



黑色像素处输出“*”，白色像素处输出“ ”(空格)。

并且这些横幅的大小还不同，具体对于大小的要求将通过给出单个像素大小 n ，即每个像素由 $n * n$ 个字符构成。请你按照要求打印出对应的横幅。

Input

输入只有一个正整数，像素大小 n ， $n \leq 10$ 。

Output

像素大小为 n 的“模拟”二字。

Examples

standard input	standard output
1	请前往比赛站点下载样例输出。
2	请前往比赛站点下载样例输出。

Problem B. 大采购

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

众所周知，龙龙十分喜欢逛中关村南门出去几百米远的那个五星超市。这天龙龙拉着 yds 一起去逛五星超市，龙龙推着一个重量上限为 N 的购物车。他们在五星超市中逛了一圈也没看见什么想买的，购物车里还是空的，这时龙龙突然发现后门附近还有一条长长的货物架，上面有 M 个商品，每个商品分别有重量 G ，和价值 S 两个属性，但这个购物架是单向的，也就是说龙龙只能从现在所处的这端走到另一端将商品按顺序放入购物车中。龙龙说他可以轻松地将一些商品放入放入购物车中使得在购物车的重量承受范围内，商品的总价值最大。但 yds 心想，这不就是一个简单的 01 背包问题？恶毒的 yds 并不想让龙龙简单的解决这个问题，他要求龙龙在只能按顺序放入商品之外，每次放入的商品的重量还必须小于等于上一次放入的商品的重量。龙龙觉得这个问题就有点费脑筋了，他并不想去仔细思考，请你帮他算出他在这样的规则下最多能往购物车里塞价值为多少的商品。

Input

第一行输入两个整数 N, M ，分别代表购物车的重量限制与商品的个数 ($1 \leq N \leq 100, 1 \leq M \leq 100$)

接下来 M 行，每行输入两个整数 G_i, S_i ，分别代表第 i 个商品的重量与价值 ($1 \leq G_i, S_i \leq 100$)

Output

输出一个整数 Sum ，表示龙龙能够放入购物车中的做大的总价值数量。

Example

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

Problem C. Changing Game

Input file: standard input
Output file: standard output
Time limit: 7 seconds
Memory limit: 256 megabytes

Welcome to *Changing Game*! In this game, you are given an integer array A of n **distinct** values $a_1, a_2, \dots, a_n$.

You are allowed to do some operations on array A . In each operation you can select an element from A and replace it with an integer c . Note that you need to keep all elements of A distinct after each operation.

Now, you are given an integer array $B = [b_1, b_2, \dots, b_n]$ which derives from A through some number of operations by Master Yi. Could you restore the operation sequence performed by Master Yi that produces B from A ? It sounds too easy to do this job. In order to increase the difficulty of this problem, Master Yi recommended that you need to find the most optimal one. In other words, you should find the minimum number of operation(s) that produces B from A .

Input

You need to process t test cases.

The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases. Then descriptions of t test cases follow.

- The first line of each test case contains one integer n ($1 \leq n \leq 100\,000$) — the length of the given arrays A and B .
- The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 1\,000\,000\,000$) — the elements of array A in order. It's guaranteed that all elements in A are distinct.
- The third line contains n integers $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq 1\,000\,000\,000$) — the elements of array B in order. It's guaranteed that all elements in B are distinct.

It is guaranteed that the sum of the length n for all test cases does not exceed 2 000 000.

Output

For each test case, you should print several lines as follow.

- The first line of each test case should contain one integer m — the minimum number of operation(s).
- Then the next m line(s), each line should contain 2 integers p ($1 \leq p \leq n$) and c ($1 \leq c \leq 1\,000\,000\,000$) — to choose the position p_i to modify $a_{p_i} \leftarrow c_i$ in i -th operation.

If there are multiple answers, you can print any of them. **Note there can't be any extra space in your output**, or you may get **Wrong Answer** verdict.

Example

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

Problem D. 欧拉树

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

欢迎来到真是强老师的数论课，今天强老师讲到了著名的欧拉函数，“对于给定的正整数 n ，它的欧拉函数 $\varphi(n)$ 是小于或等于 n 的正整数中与 n 互质的数的个数。例如，与 1 互质的正整数只有它本身，因此 $\varphi(1) = 1$ ；与 6 互质的数有 1, 5，所以 $\varphi(6) = 2$ ，大家明白了吗？”，强老师如是说。

你点了点头，表示学会了！于是，强老师决定考考你：给定一棵由 n 个节点构成的彩色无向树，每个节点都被强老师染上了一种颜色，其中第 i 个节点的颜色为 c_i 。强老师想知道，由节点 p 到节点 q 的路径上所包含的节点颜色乘积的欧拉函数是多少？换句话说，若从节点 p 到节点 q 的路径上经过的节点分别为 $a_1, a_2, \dots, a_k$ ，其中 k 为路径长度，且 $a_1 = p, a_k = q$ ，那么强老师想知道 $\varphi(\prod_{i=1}^k c_{a_i})$ 的结果是多少，答案可能很大，请取模 998244353。

Input

输入共 $n + 1$ 行，第一行输入三个正整数 n, p, q ($1 \leq n \leq 100\,000, 1 \leq p, q \leq n$) 由空格间隔开，分别表示彩色无向树的节点数和强老师想知道的点对编号 p, q 。

第二行输入 n 个正整数 $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq 1\,000\,000$) 由空格间隔开，表示每个节点所染的颜色。

接下来 $n - 1$ 行，每行输入两个正整数 u 和 v ($1 \leq u, v \leq n$)，表示彩色树中由节点 u 和 v 构成的边集 (u, v) 。

Output

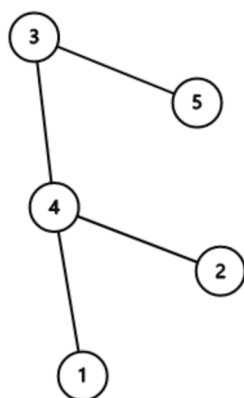
请输出一个非负整数，表示强老师想知道的答案取模 998244353 的结果。

Example

standard input	standard output
5 3 2 6 4 2 5 7 1 4 3 4 3 5 4 2	16

Note

对于样例，由节点 3 到节点 2 的路径为 $\{3, 4, 2\}$ ， $\varphi(c_3 \cdot c_4 \cdot c_2) = \varphi(40) = 16$ 。



Problem E. 简单的计算几何

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

这是一道简单的计算几何题。

给定平面上 n 个整点，其中任意三点不共线。请从中选出两个点，使其所在的直线可以把平面正好分成两个部分，使得每部分都正好包含 $\lfloor \frac{n-2}{2} \rfloor$ 个点。

Input

输入第一行为一个整数 $n(2 \leq n \leq 10^5)$ ，表示平面上点的数量。

接下来 n 行，每行两个整数 $x_i, y_i(0 \leq |x_i|, |y_i| \leq 10^9)$ ，表示第 i 个点的横纵坐标 ($i \in \{1, 2, \dots, n\}$)。

Output

输出两个整数，如果平面上不存在这样的两个点，则输出“-1 -1”。否则请输出两个整数，表示你选中的两个点的编号。

对于有解的情况，答案可能有很多种，输出任意一种即可。

Example

standard input	standard output
4 1 0 -1 0 0 1 0 -1	4 3

Problem F. Ironhead 的推免资格（传说）

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

二、拟推荐名单

序号	学号	姓名	课程成绩本 专业名次	综合成 绩	综合排 名名次	专业	思想品德 考核情况	推免情况
1	1120170001	ironhead	1	100.0	1	吉奥马丘炼铜系	合格	获推免资格（传说）
2			2	99.6	2	计算机科学与技术	合格	获推免资格（普通）
3			4	99.6	2	计算机科学与技术	合格	获推免资格（普通）
4			6	98.8	4	计算机科学与技术	合格	获推免资格（普通）
5			5	98.4	5	计算机科学与技术	合格	获推免资格（普通）
6			8	98.0	6	计算机科学与技术	合格	获推免资格（普通）
7			9	97.6	7	计算机科学与技术	合格	获推免资格（普通）
8			3	97.2	8	计算机科学与技术	合格	获推免资格（普通）
9			7	96.8	9	计算机科学与技术	合格	获推免资格（普通）
10			11	96.4	10	计算机科学与技术	合格	获推免资格（普通）

Ironhead，传说中 BIT 大陆的吉奥马丘炼铜系 yyds，以绝对优势一骑绝尘，获得推免资格（传说）。为了联合每一个保研人的魔法能力，在竖炬劫沟上建立起 $P=NP$ 大桥，ironhead 需要知道最后一位保研人的名字。然而 ironhead 很忙，没有时间来统计谁是最后一名保研人。

吉奥马丘炼铜系的保研排名计算如下。在学习成绩专业排名名次前 30%（向上取整）以内的同学中，按照综合排名从小到大逐一进行保研，直到用完保研名额。 $\text{综合排名} = \text{学习成绩专业排名名次} \times 0.85 + \text{其他表现排名名次} \times 0.15$ 。吉奥马丘炼铜系的保研名额 = 总人数 $\times 18.6\%$ ，之后向上取整。如果保研的最后名额中有人并列，那么他们的综合排名名次按姓名字典序从小到大排列。ironhead 只需要知道最后一位保研人的名字，所以有同名的也没有关系。

现在请你来计算保研结果，找出最后一名保研人。

Input

第一行一个整数 n , ($1 \leq n \leq 1000$)，表示吉奥马丘炼铜系的人数。

接下来 n 行，每行一个仅由小写英文字母组成的姓名 s ，长度不超过 100，和该同学的学习成绩名次 a 和其他表现名次 b ，且 $1 \leq a, b \leq n$ ，可能会有并列的排名名次，保证名次合法。

Output

一行一个字符串，为最后一位保研人的名字。

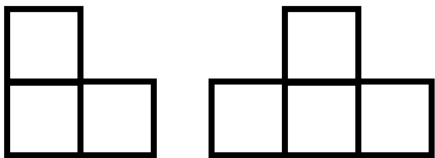
Example

standard input	standard output
1 ironhead 1 1	ironhead

Problem G. 题目册就是给你剪拼图的？

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

这天龙龙又很快的 AK 了比赛，但是很可惜，他的队友们都抛弃他去上厕所而一去不回了（可能在厕所击剑）。所以龙龙需要一个人孤独地等待四个半小时后的比赛结束。龙龙想，反正比赛都已经 AK 了，题目册已经没有用了，不如剪成一块一块的来玩拼图游戏。但龙龙有一些心 (da) 理 (nao) 变 (tan) 态 (huan)，他只想拼由下面两种拼图块组成的拼图。于是龙龙把你抓住，希望你把题目剪成他希望的样子来让他消耗剩余的时间。



为了简化问题，我们认为拼图块里的每一个小格都是 $1cm \times 1cm$ 的，并且题目册的长宽也都是整数的规整的矩形。

Input

输入两个正整数 N, M ，分别表示题目册短边和长边的长度 ($1 \leq N \leq M \leq 100$)

Output

没有合法的裁剪方案输出 “-1”，有多种方案，输出任意一种方案即可，输出方式如下：

输出 N 行，每行 M 个数，请对每块拼图块从 1 开始编号，按编号输出，例子见输出样例。**注意行末不要输出多余的空格，注意换行。**否则你可能会得到 Wrong Answer。

Examples

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

Problem H. Prefix Sum

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 256 megabytes

We define

$$a_n^{(k)} = \sum_{i=1}^n a_i^{(k-1)}, k \geq 1$$

You are given the initial sequence $a^{(0)}$, k and several queries, each query satisfies one of the following format:

- **1 p q**: add q to $a_p^{(0)}$
- **2 l r**: ask the value of $\sum_{i=l}^r a_i^{(k)}$

you have to answer the second type queries, each in one line. Since this value may huge, you should output this value *mod* 998244353.

Input

The first line contains two integers, the length of the sequence n , and k ($1 \leq n, k \leq 10^4$).

The second line contains n integers, $a_1^{(0)}, a_2^{(0)}, \dots, a_n^{(0)}$ ($0 \leq a_i^{(0)} < 998244353, i = 1, 2, \dots, n$).

The third line contains an integer m ($1 \leq m \leq 2 \times 10^5$), the number of queries.

The next m lines, each line contains 3 integers “1 p q” ($1 \leq p \leq n, 0 \leq q < 998244353$) or “2 l r” ($1 \leq l \leq r \leq n$), represent the queries.

It is guaranteed that there is at least one “2 l r” query.

Output

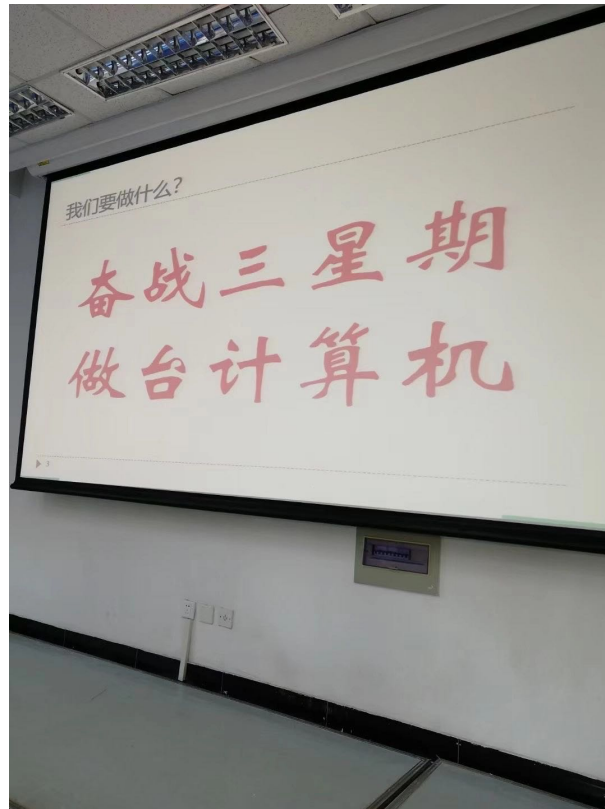
For each “2 l r” query, output an integer, represent the answer, each in one line.

Example

standard input	standard output
3 2	17
1 3 2	20
4	1
2 1 3	
1 2 1	
2 1 3	
2 1 1	

Problem I. shenyunhan Loves Palindrome

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



奋战三星期，造台计算机。

shenyunhan has made a computer. This computer is special, it can determine whether a substring of some template string is a **cyclic palindrome**.

Now shenyunhan wants to verify his computer. shenyunhan will give you a template string s and some questions. For each question, shenyunhan wants to know whether some substring of s is cyclic palindrome.

A string s of length n is **palindrome** if and only if the following condition holds:

$$(\forall i \in [1, n]) (s[i] = s[n - i + 1])$$

A string s of length n is **cyclic palindrome** if and only if the following condition holds:

$$(\exists N \in [n, 2n - 1]) ((\forall i \in [1, N]) (s[(i - 1) \bmod n + 1] = s[(N - i) \bmod n + 1]))$$

Note that string indexes starts from 1.

It's obvious that all palindromes are cyclic palindromes.

Input

The first line of input contains two space separated integers n and q ($1 \leq n, q \leq 2 \times 10^5$). n is the length of the template string and q is the number of questions shenyunhan is going to ask you.

The next line contains the template string s of length n . It's guaranteed that all characters in s are lowercase English letters.

For the next q lines, the i -th line contains two space separated integer l_i and r_i ($1 \leq l_i \leq r_i \leq n$) which represents the i -th question that shenyunhan asks you. Your program needs to figure out whether the substring $s[l_i..r_i]$ is a cyclic palindrome.

The notation $s[L..R]$ represents the substring of s that spans from the character at index L to the character at index R , inclusive. For example, `hello[2..4] = ell`. Note that string indexes start from 1.

Output

For each question asked by shenyunhan, please output the answer on a separate line. For the i -th question, if the substring $s[l_i..r_i]$ is a cyclic palindrome, output YES; otherwise output NO.

The judge is case-insensitive, so you can output your answer in any case you wish.

Example

standard input	standard output
10 6	NO
bbccbcbbac	NO
8 10	YES
1 10	YES
9 9	YES
9 10	NO
4 6	
1 7	

Note

In the given example test case:

- For the first query, $s[8..10] = \text{bac}$, which is not a cyclic palindrome;
- For the second query, $s[1..10] = \text{bbccbcbbac}$, which is not a cyclic palindrome;
- For the third query, $s[9..9] = \text{a}$, which is a cyclic palindrome since it's a palindrome;
- For the forth query, $s[9..10] = \text{ac}$, which is a cyclic palindrome;
- For the fifth query, $s[4..6] = \text{cbc}$, which is a cyclic palindrome since it's a palindrome;
- For the sixth query, $s[1..7] = \text{bbccbc}$, which is not a cyclic palindrome.

Problem J. 屠龙勇者 ErvinXie

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

又是一个俗套的故事，吉奥马丘国的公主被恶龙抓走了。作为吉奥马丘国最出名的勇者，*ErvinXie* 理所当然的担负起了拯救公主，成为屠龙勇士的责任。*ErvinXie* 极其强大，他攻无不克、战无不胜，对付恶龙简直就是以大欺小，恃强凌弱、虎入羊群、泰山压卵、席卷八方、翻手为云、覆手为雨。所以他击败恶龙完全不需要什么准备。但是，传说屠龙勇士终将成为恶龙。*ErvinXie* 为了自己下半生的幸福，当然不能容忍自己成为一条恶龙。所以，*ErvinXie* 从吉奥马丘国的炼金大师 *XIRHXQ* 那里学习了驱魔法阵的制作方法，同时炼金大师 *XIRHXQ* 好心的给 *ErvinXie* 提供了两套一次性传送阵法。*ErvinXie* 为了凑齐驱魔法阵的布阵材料，需要前往竖炬劫沟旁的季蒜脊河里采集材料。季蒜脊河是一条神奇的河，它里面藏着很多奥秘。季蒜脊河里每一米都有且仅有着一个特殊的炼金材料。为了能够成功布置驱魔法阵，同时尽快救出公主。*ErvinXie* 打算使用一次性传送阵法传送到季蒜脊河中某一个位置（*ErvinXie* 可以指定是哪一個位置），然后在河中收集炼金材料。收集完材料后，立马传送去恶龙谷拯救公主。

已知这个世界中一共有 k 种炼金材料，分别使用 $1, 2, 3 \dots k$ 编号。

炼金法阵的布置方法是将相应炼金材料按直线摆放就能成功布置。法阵一共需要 s 个炼金材料，法阵的布置方法通过 $a_1, a_2, a_3 \dots a_s$ 给出，其中 a_i 代表的是炼金材料种类。可能存在不同位置需要摆放相同种类炼金材料的情况。那么这种材料就需要收集多个。

季蒜脊河的长度为 len ，从源头开始直到末尾，每米存在的炼金材料为 $b_1, b_2, b_3 \dots b_{len}$ 。 b_i 是炼金材料的种类。

问 *ErvinXie* 至少需要经过多少个炼金材料（包含收集的）才能去屠龙拯救公主。当然，如果 *ErvinXie* 注定收集不全材料，他仍然会去拯救公主，和公主在一起... 即使是以恶龙的身份。

例如：季蒜脊河 5 米长，材料为 $[1, 2, 1, 3, 2]$ ，法阵的布置方法为 $[1, 2, 3, 2]$ ，那么 *ErvinXie* 可以直接传送到河里的第 2 个位置，往后收集材料，最终按照顺序收集了 $[2, 1, 3, 2]$ 材料后，能够完成法阵，此时传送去恶龙谷拯救公主。经过了 4 个炼金材料。

PS：不会吧不会吧，你不会认为 *ErvinXie* 会按收集顺序摆放法阵吧？*ErvinXie* 只是懒，不是蠢。他当然知道材料收集够了，摆放可以自己按顺序摆。

Input

第一行一个数字 k ，代表这个世界中存在的炼金材料种类数。 $1 \leq k \leq 5000$

第二行一个数字 s ，代表法阵的长度。 $1 \leq s \leq 10^6$

第三行 s 个数字，代表法阵的布置方式 a_i 。 $1 \leq a_i \leq k$

第四行一个数字 len ，代表季蒜脊河的长度。 $1 \leq len \leq 2 \times 10^6$

第五行 len 个数字 b_i ，代表季蒜脊河每米存在的炼金材料的种类。 $1 \leq b_i \leq k$

Output

输出为一行

如果能够收集完材料，输出一个数字，表示 *ErvinXie* 至少需要经过多少个炼金材料（包含收集的）才能去屠龙拯救公主。

否则... 输出 “DragonXie” （不含引号）

Examples

standard input	standard output
5 4 1 2 3 2 5 1 2 1 3 2	4
5000 1 512 5 1 2 3 4 5	DragonXie

Note

输入数据较大，推荐使用 `scanf` 或者自行实现快速读

Problem K. Strange Game

Input file: standard input
Output file: standard output
Time limit: 3.5 seconds
Memory limit: 512 megabytes

shenyunhan is playing a strange game with his best friend. The game is based on a directed graph. Each node in the graph contains an integer which represents the “score” of the node. When playing the game, the two players take turns removing edges from the graph. On each turn, each player can only remove exactly one edge. After some player finishing removing some edge, all nodes that is not reachable from a special node named “origin node” are removed from the graph and the player removing the edge gets a score that is the sum of all scores on these removed nodes. Note that when a node is removed from the graph, all edges associated with the node is also removed.

shenyunhan did not sleep well last night, so he decided to get some help from you. Specifically, shenyunhan will give you a number of questions. In each question, shenyunhan will ask you the score he can get if he removes some edge from the graph in the current turn. Can you help poor shenyunhan?

Node T is reachable from node S if and only if one of the following two conditions is true:

- There is an edge from S to T present in the graph;
- There is a non-empty sequence of nodes $I_1, I_2, I_3, \dots, I_k$ such that $S \rightarrow I_1, I_1 \rightarrow I_2, I_2 \rightarrow I_3, \dots, I_k \rightarrow T$ are all present edges in the graph.

Input

The first line contains two space separated integers N and M ($1 \leq N, M \leq 4 \times 10^5$) representing the number of nodes and the number of edges in the graph. Nodes in the graph are numbered from 1 to N and edges in the graph are numbered from 1 to M . Node 1 is the “origin node”.

Then there are one line containing N space separated integers s_i ($1 \leq i \leq N, 1 \leq s_i < 2^{31}$). The i -th of these integers represents the score of the i -th node.

Then there are M lines of input, the i -th ($1 \leq i \leq M$) of which contains two separated integers u_i and v_i ($1 \leq u_i, v_i \leq N$) representing the i -th edge in the graph that spans from node u_i to node v_i . It is guaranteed that initially all nodes in the graph are reachable from the “origin node”.

Then the input contains a single integer Q ($1 \leq Q \leq 10^5$) representing the number of questions shenyunhan asks you.

Then there are Q lines of input, each of which contains a single integer q_i ($1 \leq q_i \leq M$) which is the i -th question asked by shenyunhan. For each question, shenyunhan wants to know how many scores he can get if he removes the q_i -th edge from the graph in the current turn.

Output

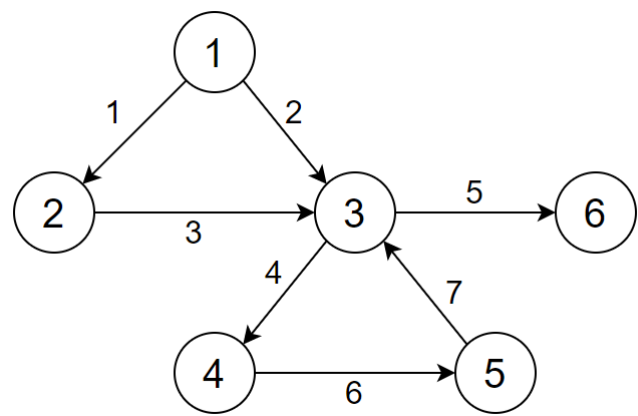
For each question asked by shenyunhan, output a single line containing a single integer that represents the answer to the question.

Example

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

Note

The graph given in the example test case is visualized in the following figure.



If shenyunhan removes the 1st edge which spans from node 1 to node 2, then node 2 is unreachable from node 1 so shenyunhan get score 2.

If shenyunhan removes the 2nd edge which spans from node 1 to node 3, no nodes becomes unreachable from node 1 so shenyunhan get a zero score.

Other answers of the given queries are figured out just like above.

Problem L. “我为什么要升级？”

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

龙龙最近喜欢上了玩一些资源收集类的挂机游戏。这类挂机游戏有个基本模型就是，对于资源产出单位的升级，升级的资源消耗是指数型上升的，但资源产出的提升是线性的。我们将这个模型简化成下面的样子：

对于一个产出金币的建筑，第 1 次升级消耗 N_1 个金币，每次升级消耗金币的提升比例为 $P(N_{i+1} = N_i \times P)$ ，每次升级后，每小时的金币产出的增加值 M 都是固定的。从一方面看，仅有升级后增加的金币产量才属于本次升级的收益，因此我们可以定义一个“金币回收时长”作为我们升级该建筑的利益的一个衡量标准，其值等于当次升级消耗的金币量除以每小时增加的产出金币量。例如第 3 次升级该建筑时，金币回收时长为 $(N_1 \times P \times P) \div M$ 个小时。

龙龙常常一算发现他需要几百个小时才能收回他的升级消耗的金币，总是边感慨“我为什么要升级？”边默默点下升级按钮。他已经不想再经受这种打击了，所以不会再去算金币回收时长，但恶毒的yds想指派你算出来时间告诉龙龙，来打击他的心态。

Input

第一行输入一个整数 T ，代表样例的组数 ($1 \leq T \leq 100$)

接下来 T 行，每行输入四个数 N_1, P, M, Q ，分别代表第 1 次升级时的消耗金币数，每次升级消耗金币数的提升比例，每次升级后金币产量的提升量，和龙龙现在要进行的是第 Q 次升级，其中 N_1, M, Q 都是整数， P 是有理数。 ($1 \leq N_1, M \leq 1000, 1 \leq P \leq 1.3, 1 \leq Q \leq 50$)

Output

yds 并不打算费尽口舌说一个很长的数字，所以请使用%.1f 来使输出仅保留 1 位小数，表示本次升级的金币回收时长。

Example

standard input	standard output
3	61.2
1000 1.1 100 20	319.5
1000 1.2 100 20	1067.2
1000 1.1 100 50	

Note

注意浮点精度，请使用 double 类型进行运算。

Problem M. 这局我觉得你能赢

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

老阴阳人鲍勃在吉奥马丘国经营着一家酒馆。每天晚上，吉奥马丘国的玩家们都会来酒馆里下酒馆战棋。酒馆战棋是一种风靡吉奥马丘国的桌游，对战双方会召唤一系列随从，按顺序随机相互进攻，直到一方的随从被完全消灭为止。有一些随从有特殊能力，比如攻击特效或者死亡特效。

一天，ironhead 和 XIRHXQ 来到酒馆里下棋，他们都是鲍勃最喜欢的指挥官，所以鲍勃都会暗中帮助他们。鲍勃会在对战开始之前，悄悄告诉赢面大的人“这局我觉得你能赢”。这次轮到他们俩对战了，鲍勃不愿意暗示他们谁会输，所以鲍勃决定对赢面小的人说“我有种预感，我觉得这次你能赢”。你作为鲍勃的助手，请你按照以下规则和方法计算出谁的赢面大。



1. 基本规则：

(1) 随从：任意一个随从都有生命值和攻击力。当两个随从交战时，互相对对方造成自己攻击力的伤害，然后用生命值减去此伤害。当一个随从的生命值小于等于 0 时，它就立即死亡。随从属于不同的种族，在本题中，只有机械和非机械两种。每个随从都有特殊的效果，在特定条件下触发。

(2) 对战场地：双方场地内最多各有 7 个随从。在召唤时，如果已经有 7 个随从，就停止后续召唤。

(3) 对战规则：双方随从从左至右交替进行攻击，直到某一方的随从全部死亡且没有还未结算的特殊效果。当最右边的随从攻击后，从最左边的随从开始新一轮攻击。随从的攻击目标是随机的。先攻方也是随机的。最后有随从存活的一方胜利，如果双方都没有随从存活，则为平局。

(4) 行动顺序：从左至右交替攻击，即每一方从左至右依次派出随从攻击一次，一次攻击完成后交换攻击方。当某个随从例如 A，攻击完成后，不论 A 有没有死亡，下一次本方进攻的随从是 A 右边的随从 B（当 A 为最后一个随从，那么就是第一个随从；如果 A 召唤了新随从，那就是新随从）。如果随从 B 还没有来得及进攻就被打死了，那么就由 B 右边的随从 C 进攻。

2. 特殊效果：

(1) 光环：当该随从在场上时，始终触发的效果，同名光环效果可以叠加。

(2) 亡语：随从的效果，在随从死亡后触发。

(3) 召唤：从自己右边，召唤目标随从。

3. 随从列表:

名称	攻击力	生命值	种族	效果
机械蛋	1	5	机械	亡语：召唤一个机械龙。
机械龙	4	4	机械	无特殊效果。
机械学徒	3	6	无	亡语：依次召唤本局最早死亡的两个己方机械随从。(若没有死亡的机械随从，就不召唤。只死亡一个，就召唤死亡的那一个)。
卡德加	2	2	无	光环：己方召唤的随从翻倍。

注意:

亡语召唤和卡德加：一个机械蛋死亡后，召唤 1 个机械龙；如果己方有 1 个卡德加，那么召唤 2 个机械龙；2 个卡德加则召唤 4 只机械龙，以此类推。机械学徒死亡召唤机械时，先召唤一个最先死亡的随从 A ，再在 A 右边召唤一个第二个死亡的随从 B ，即召唤了 AB ；如果有一个卡德加则先召唤 2 个随从 A ，再在 A 右边召唤 2 个随从 B ，相当于召唤了 $AABB$ ，以此类推。

4. 鲍勃的计算方式:

模拟计算 10 000 次对战，根据模拟结果给出预测。

随机：所有的随机均由以下随机函数给出，每一次随机时，按顺序调用此函数，并且取模获得相应的结果。

```
unsigned seed;

unsigned myrand(unsigned m) {
    seed = seed * 22695477 + 1;
    return seed%m;
}
```

只有两种情况需要随机。

随机情况一：战斗开始前判断谁先手；随机数范围： $x = [0, 2)$ ，即 $m = 2$ ；随机方法： $x = 0$ 则 ironhead 先手， $x = 1$ 则 XIRHXQ 先手。

随机情况二：随从攻击目标的选择；随机数范围： $x = [0, n)$ ， n 为对方当前随从个数，即 $m = n$ ；随机方法：攻击从左至右第 $x + 1$ 个随从。

在模拟一场战斗时，先调用随机函数判断先手，随后按照随从的攻击顺序依次调用随机函数得出攻击目标。

Input

总共三行输入

第一行一个 32 位无符号整数 seed，代表最初的 seed 值。

第二行首先一个数 $n (0 \leq n \leq 7)$ ，代表 ironhead 的随从数目，接下来 n 个整数代表从左至右的随从 (0: 机械蛋, 1: 机械龙, 2: 机械学徒, 3: 卡德加)。

第三行同第二行，但是代表 XIRHXQ 的随从。

Output

一行一个整数，代表模拟计算中 ironhead 胜利的场次。

Examples

standard input	standard output
1 1 1 0	10000
123456 3 0 0 3 2 0 2	4990