Problem A. 制表符

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

在编写代码的时候，有些人习惯使用制表符（Tab 键）来控制缩进、作分隔等，而有一些人则喜欢使用空格来做这些事情。

在一些代码编辑器中，有一种将制表符转化为空格的功能。使用者只需设定好每个制表符要用多少个空格来替代，代码编辑器就会自动地对代码进行文本替代，使其代码风格与使用者接近。

这里将会给出一份 n 行的代码，你的任务是写一个程序，将代码中的所有制表符（用 "->" 表示）转化为 k 个空格，并将转化后的代码输出。

Input

第一行包含一个正整数 $T(1 \le T \le 10)$，表示测试数据的组数。

每组测试数据第一行包含两个正整数 $n, k(1 \le n \le 10, 1 \le k \le 8)$，分别表示代码的行数以及制表符应该被替换成多少个空格。

接下来 n 行，每行一个非空的字符串，表示每行代码。每个字符串长度均不超过 50，且字符串仅由小写字母 "a" 到 "z"、数字 "0" 到 "9"、特殊字符 "{","}","(",")",";","-",">" 构成。

Output

对于每组测试数据，输出 n 行，每行一个字符串，表示每行代码转化后的结果。

Examples

stdin	stdout

2
5 2
dfs(x){
->if(x){
->->dfs(x);
->}
}
1 3
->->->print(a);->print(b);
	dfs(x){
	  if(x){
	    dfs(x);
	  }
	}
	          print(a); print(b);

Problem B. 序列归并

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

Alice 和 Bob 正在对两个序列 $a_1, a_2, \ldots, a_n$ 和 $b_1, b_2, \ldots, b_m$ 进行操作。

Alice 首先需要将它们归并成一个长度为 $n+m$ 的序列 $c_1, c_2, \ldots, c_{n+m}$。即将序列 a 和 b 合并成一个序列 c，但不改变 a 和 b 的顺序。显然可能有许多许多种不同的归并结果。

Bob 需要在 Alice 完成归并之后，选择一对 l, r，满足 $1 \le l \le r \le n+m$，并使得 $score = c_l + c_{l+1} + c_{l+2} + \cdots + c_{r-1} + c_r$ 尽可能地大。

不同的归并结果以及不同的选择 l, r 的方式都会使得最后 $score$ 的值不尽相同，请问 $score$ 最多能达到多少呢？

Input

第一行包含一个正整数 $T(1 \le T \le 10)$，表示测试数据的组数。

每组测试数据第一行包含两个正整数 $n, m(1 \le n, m \le 100000)$，分别表示序列 a 和序列 b 的长度。

第二行包含 n 个整数 $a_1, a_2, \ldots, a_n(-10^9 \le a_i \le 10^9)$。

第三行包含 m 个整数 $b_1, b_2, \ldots, b_m(-10^9 \le b_i \le 10^9)$。

输入数据保证 a 和 b 中至少有一个正数。

Output

对于每组测试数据，输出一行一个整数，即 $score$ 的最大值。

Examples

stdin	stdout
1 2 3 1 5 3 -1 -1	9

Notes

一种可能的归并结果是 $c = [1, 3, 5, -1, -1]$，选择 $l = 1, r = 3$，则 $score = c_1 + c_2 + c_3 = 9$。

Problem C. Least Common Multiple

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

In arithmetic and number theory, the least common multiple of two integers a and b, usually denoted by $LCM(a, b)$, is the smallest positive integer that is divisible by both a and b.

The LCM of more than two integers is also well-defined: it is the smallest positive integer that is divisible by each of them.

You are given n positive integers $a_1, a_2, \ldots, a_n$, please find the LCM of them.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10)$, denoting the number of test cases.

In each test case, there is one integer $n(2 \leq n \leq 15)$ in the first line.

In the second line, there are n integers $a_1, a_2, ..., a_n(1 \leq a_i \leq 10^{18})$.

Output

For each test case, print a single line containing an integer, denoting the LCM. Since the answer can be extremely large, please print it modulo $10^9 + 7$ instead.

Examples

stdin	stdout
3 2 4 6 3 7 8 9 2 1000000009 1000000009	12 504 2

Problem D. Mo's Algorithm

Input file:	`stdin`
Output file:	`stdout`
Memory limit:	256 megabytes

Mo's algorithm can solve some difficult data structure problems. Like this: You are given an array $a_1, a_2, \ldots, a_n$ and m queries. In each query, we want to get the mode number(the value that appears most often in a set of data) of subsequence $a_l, a_{l+1}, \ldots, a_r$.

Maybe you want to solve it using segment tree or some other data structures. However, if you know the answer of $[l, k]$ and $[k+1, r]$, you will find that it is very difficult to get the answer of $[l, r]$.

Mo's algorithm is an offline algorithm that can solve this problem efficiently.

First, we need a data structure that can do these operations:

- Insert a number.
- Delete a number.
- Get the mode number.

It can be solved easily by a binary heap or a balanced binary search tree. We call this data structure DS.

On the other hand, if we only keep $a_l, a_{l+1}, \ldots, a_r$ in DS, we can transform to $[l', r']$ in $|l-l'|+|r-r'|$ moves. So we need to find a good order to perform queries that the total number of moves is minimized.

It is very hard to find such excellent order. Mo's algorithm just gives an order that the total number of moves is $O(m\sqrt{n})$.

In this task, you are given the length of the sequence n and m queries $q_1, q_2, \ldots, q_m$. Please write a program to rearrange these queries that $cost(q_1, q_2) + cost(q_2, q_3) + \cdots + cost(q_{m-1}, q_m)$ is minimized, where $cost([l, r], [l', r']) = |l - l'| + |r - r'|$.

Input

The first line of the input contains an integer $T(1 \le T \le 10)$, denoting the number of test cases.

In each test case, there are two integers $n, m(1 \le n \le 100000, 2 \le m \le 10)$ in the first line, denoting the length of the sequence and the number of queries.

For the next m lines, each line contains two integers l_i and $r_i(1 \le l_i \le r_i \le n)$, denoting a query.

Output

For each test case, print a single line containing an integer, denoting the minimum number of moves.

Examples

stdin	stdout
2 5 2 1 5 3 4 9 5 5 5 1 1 4 4 2 2 3 3	3 8

Problem E. 最近距离

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

在平面上有 n 个边平行坐标轴的正方形，编号依次为 $1, 2, \ldots, n$，每个正方形都占据了若干个格子。第 i 个正方形的中心位于格子 (x_i, y_i)，其半径为 r_i，即它的左下角为格子 $(x_i - r_i, y_i - r_i)$，右上角为格子 $(x_i + r_i, y_i + r_i)$，共占据 $(2r_i + 1)^2$ 个格子。

定义正方形 i 和正方形 j 的距离为：从正方形 i 占据的格子中选择一个格子 (x_1, y_1)，从正方形 j 占据的格子中选择一个格子 (x_2, y_2)，使得这两个格子的曼哈顿距离 $|x_1 - x_2| + |y_1 - y_2|$ 最小，此时这两个代表格子之间的曼哈顿距离被视为这两个正方形的距离。

请写一个程序，快速支持 m 次操作，操作有以下两种：

- `1 i x y r` 将第 i 个正方形的中心改为 (x, y)，半径改为 r。
- `2 u v` 询问如果在编号在 $[u, v]$ 之间的所有正方形之中挑选两个编号不同的正方形，那么它们的距离的最小值是多少。

Input

第一行包含一个正整数 $T(1 \le T \le 3)$，表示测试数据的组数。

每组测试数据第一行包含两个正整数 $n, m(2 \le n, m \le 100000)$，分别表示正方形的数量以及操作的数量。

接下来 n 行，每行三个正整数 $x_i, y_i, r_i(1 \le x_i, y_i, r_i \le 10)$，依次描述每个正方形。

接下来 m 行，每行描述一个操作，其中 $1 \le i \le n, 1 \le x, y, r \le 10, 1 \le u < v \le n$。

Output

对于每个询问，输出一行一个整数，即距离的最小值。

Examples

stdin	stdout
1 5 5 9 3 1 2 9 1 1 2 1 8 7 1 3 7 1 2 4 5 1 2 5 6 1 2 2 4 1 3 9 5 1 2 2 4	3 1 0

Problem F. Split the Number

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

There is an infinite integer sequence $f_1, f_2, f_3, \ldots$, where $f_1 = 1, f_2 = 2, f_i = f_{i-1} + f_{i-2} (i \geq 3)$.

You are given two positive integers n and k, your task is to check whether n can be split into k integers $a_1, a_2, \ldots, a_k$, where $a_1, a_2, \ldots, a_k \in f$ and $a_1 + a_2 + \cdots + a_k = n$. Note that you can split n into same integers, for example, $10 = 5 + 5 = f_4 + f_4$.

Input

The first line of the input contains an integer $T (1 \leq T \leq 100000)$, denoting the number of test cases.

In each test case, there is one integer $n (1 \leq n, k \leq 10^{18})$ in the first line.

Output

For each test case, print a single line. If it is possible to split n into k integers, print `Yes`, otherwise print `No`.

Examples

stdin	stdout
4 10 1 10 2 5 1 12 2	No Yes Yes No

Problem G. Tunnels

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

There are n holes and $n-1$ tunnels connecting them. The holes are labeled by $1, 2, \ldots, n$. For all $i > 1$, a hole number i is connected by a tunnel to the hole number $\lfloor \frac{i}{2} \rfloor$. Tunnels are all bidirectional.

Initially, all the tunnels are available. There will be m events happened in the tunnel system. The i-th event will change the status of the tunnel between u_i and $\lfloor \frac{u_i}{2} \rfloor$. If this tunnel is available, it will become unavailable, and if this tunnel is unavailable, it will become available.

Please write a program to support these events, and report the number of pairs $i, j (1 \le i < j \le n)$ that holes i, j are still connected after each event.

Input

The first line of the input contains an integer $T (1 \le T \le 5)$, denoting the number of test cases.

In each test case, there are two integers $n, m (2 \le n, m \le 100000)$ in the first line, denoting the number of holes and the number of events.

For the next m lines, each line contains an integer $u_i (2 \le u_i \le n)$, denoting an event.

Output

For each test case, print m lines. The i-th line contains an integer, denoting the number of connected pairs after the i-th event.

Examples

stdin	stdout
1 9 6 6 4 2 4 4 3	28 13 7 13 7 5

Problem H. 树上后缀数组

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

如果要比较两个字符串的大小，一般方法是先比较第一位的大小，当第一位相同时再比较第二位，当第一位和第二位都相同时再比较第三位 ... 以此类推，直到两个字符串的第一位不同或者两个字符串完全相同为止。一般认为空串小于任何串。

对于长度为 n 的字符串 $S[1,n]$，它有 n 个后缀，第 i 个后缀为 $S[i,n]$ 这个子串。串 S 的后缀数组为数组 $sa[1,n]$，其中 $sa[i]$ 表示将这些后缀从小到大排序后，排名第 i 小的后缀是谁。

给定一棵 n 个点的以 1 为根的有根树，每个节点上有一个小写字符 c_i。定义字符串 str_i 表示从 i 点出发一直沿着父亲往上走到根，这一路中经过的节点上写着的字符从左往右拼起来得到的字符串（包括 i 点和根节点）。

请对于这棵树，求出它的 “后缀数组”，即将 $str_1, str_2, \ldots, str_n$ 从小到大排序。特别地，如果两个串相等，那么我们认为起点编号较小的串比较小。

Input

第一行包含一个正整数 $T(1 \le T \le 10)$，表示测试数据的组数。

每组测试数据第一行包含两个正整数 $n, m(1 \le n \le 100000, 1 \le m \le 26)$，分别表示点数以及字符范围。

第二行包含一个长度为 n 的字符串 $c_1, c_2, \ldots, c_n$，依次表示每个节点上的字符。输入数据保证所有 c_i 都在前 m 小的小写字母中**完全随机生成**。

第三行包含 $n-1$ 个正整数 $f_2, f_3, \ldots, f_n(1 \le f_i < i)$，其中 f_i 表示 i 点在树上的父亲节点。

Output

对于每组数据，输出一行 n 个整数 $res_1, res_2, \ldots, res_n$，即排序结果，其中 res_i 表示排序后第 i 小的字符串是 str_{res_i}。

Examples

stdin	stdout
1 6 3 cabcca 1 2 2 3 5	2 6 3 1 4 5

Problem I. Obstacles

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

There are $n \times m$ cells on the grid, the left-top cell is at $(1, 1)$ while the right-bottom cell is at (n, m).

Alice is standing at $(1, 1)$, her goal is to reach (n, m). When she is at cell (x, y), she can next move to $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$ or $(x, y + 1)$. She can't move outside the grid, and she can't move to any cell with obstacles.

Bob wants to prevent Alice from reaching (n, m). He is now at a shop selling obstacles. There are k types of obstacles. For the i-th type, he can pay w_i dollars to make all the cells (x, y) that $xl_i \le x \le xr_i$ and $yl_i \le y \le yr_i$ filled with obstacles.

Bob doesn't know how to choose these obstacles. Please write a program to help him find the cheapest way that Alice can't reach (n, m) from $(1, 1)$, or determine it is impossible.

Input

The first line of the input contains an integer $T(1 \le T \le 10)$, denoting the number of test cases.

In each test case, there are three integers $n, m, k(2 \le n, m \le 10^9, 1 \le k \le 2000)$ in the first line, denoting the size of the grid and the number of types.

For the next k lines, each line contains five integers $xl_i, xr_i, yl_i, yr_i, w_i(1 \le xl_i \le xr_i \le n, 1 \le yl_i \le yr_i \le m, 1 \le w_i \le 10^9)$, denoting a type. It is guaranteed that no obstacle will cover $(1, 1)$ or (n, m).

Output

For each test case, print a single line containing an integer, denoting the minimum number of dollars that Bob needs to pay for. If there is no solution, please print "-1" instead.

Examples

stdin	stdout
2 5 6 4 2 5 1 2 7 1 4 4 6 8 2 2 3 3 6 3 3 3 3 5 10000 10000 1 100 100 100 100 1	20 -1

Problem J. 天赋系统

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

在天赋系统中，一共有 n 个技能，编号依次为 1 到 n，每个技能可以学习很多级，也可以不学习。

一共有 m 点技能点，第 i 个技能每往上多点一级需要消耗 c_i 点技能点。

有些技能存在前置技能，一共有 k 对前置关系，第 i 对关系为：学习技能 v_i 需要前置技能 u_i。如果 u_i 学习了 0 级，那么不能学习 v_i。一个技能可以学习当且仅当它的所有前置技能都学习了至少一级。

请对于每个 $k = 1, 2, 3, \ldots, m$ 计算，有多少种点技能的方式满足刚好消耗了 k 点技能点。两个方案不同当且仅当某个技能的学习等级在两个方案中不同。

Input

第一行包含一个正整数 $T(1 \le T \le 10)$，表示测试数据的组数。

每组测试数据第一行包含三个正整数 $n, m, k(1 \le n \le 15, 1 \le m \le 1000, 1 \le k \le \frac{n(n-1)}{2})$，分别表示技能的数量、技能点的数量以及前置关系的数量。

第二行包含 n 个正整数 $c_1, c_2, \ldots, c_n(1 \le c_i \le m)$，依次表示每个技能所需的技能点。

接下来 k 行，每行包含两个正整数 $u_i, v_i(1 \le u_i < v_i \le n)$，描述一对前置关系。输入数据保证同一对关系不会被输入多次。

Output

对于每组数据，输出一行 m 个整数 $ans_1, ans_2, \ldots, ans_m$，其中 ans_i 表示恰好消耗 i 点技能点的方案数。因为答案可能很大，请对 $10^9 + 7$ 取模输出。

Examples

stdin	stdout
1 3 8 1 2 4 3 1 3	0 1 0 2 1 2 1 4

Problem K. 二进制码

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

在计算机中，对于定点数有三种不同的表示方法。在本题中，假定码的长度固定为 8 位，从左往右依次编号为第 1 到 8 位，第 1 位为最高位。

x 的原码：最高位为符号位，正数符号位为 0，负数符号位为 1，第 2 到 7 位为 x 的二进制表示。正负 0 的原码不同。

x 的反码：原码符号位除外，其他位按位取反，即 1 变 0，0 变 1。

x 的补码：正数的补码等于原码，负数的补码等于反码 $+1$，因此正负 0 的补码相同。

给定整数 x，请给出它的原码、反码和补码。

Input

第一行包含一个正整数 $T(1 \leq T \leq 300)$，表示测试数据的组数。

每组测试数据包含一行，首先是一个符号 "+" 或 "-"，表示 x 的正负，然后是一个非负整数 $y(0 \leq y \leq 100)$，表示 x 的绝对值为 y。

Output

对于每组数据，输出三行，第一行为原码，第二行为反码，第三行为补码。

Examples

stdin	stdout
4 +0 +1 -0 -3	00000000 01111111 00000000 00000001 01111110 00000001 10000000 11111111 00000000 10000011 11111100 11111101

Problem L. 小地图

Input file:	stdin
Output file:	stdout
Memory limit:	256 megabytes

给定一个 n 行 m 列的网格地图，从上到下依次编号为第 1 行到第 n 行，从左往右依次编号为第 1 列到第 m 列。每个格子要么可以通行（用 “.” 表示），要么不可通行（用 “#” 表示）。

玩家 P 现在位于某个可以通行的格子上，小地图显示出了玩家所在位置周围 $3 \times 3 = 9$ 个格子。请写一个程序，根据地图和小地图找到玩家可能所在的位置。

Input

第一行包含一个正整数 $T(1 \leq T \leq 10)$，表示测试数据的组数。

每组测试数据第一行包含两个正整数 $n, m(1 \leq n, m \leq 100)$，表示行数和列数。

接下来 n 行，每行包含一个长度为 m 的字符串，依次描述每一行每个格子的情况。字符串仅由 “.” 和 “#” 构成。

接下来 3 行，每行包含一个长度为 3 的字符串，描述玩家所在位置周围的情况。字符串由 “.”、“#”、“P”（表示玩家所在位置）和 “*” （表示超出地图的部分）构成。输入数据保证有且仅有一个 “P”，且 “P” 一定位于小地图的第 2 行第 2 列。

请注意：你不能旋转/翻转小地图。

Output

对于每组数据，输出一行一个整数，即玩家可能所在的位置的数量。

Examples

stdin	stdout
2 5 8 #.#....# #.#..#.. .#..#.#. #..###.. ...#.... .#. #P. ... 1 2 .# *** *P# ***	2 1