Problem A. Calculus

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

This summer, ZXyang became so tired when doing the problems of Multi-University contests. So he decided to attend the Unified National Graduate Entrance Examination. This day, he sees a problem of series.

Let $S(x)$ be a function with x as the independent variable. $S(x)$ can be represented by the formula as follow.

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

$$S(x) = \sum_{j=1}^{x} f(j)$$

$f_i(x)$ is a function with x as the independent variable. Furthermore. $f_i(x)$ belongs to the function set F.

$$F = \left\{ C, \frac{C}{x}, C\sin x, C\cos x, \frac{C}{\sin x}, \frac{C}{\cos x}, Cx, C^x \right\}$$

C is a constant integer ranging from 0 to 10^9.

ZXyang wonders if $S(x)$ is convergent. $S(x)$ is convergent if and only if $\lim_{x\to\infty} S(x) = c$, where c is a constant.

Input

The first line of input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first and the only line of each test case contains a single string s ($1 \le |s| \le 100$), indicating the formula of $f(x)$. Fraction is presented as a/b. C^x is presented as `C^x`. It's guaranteed that the constant C won't be left out when $C = 1$. $f(x)$ consists of functions from F connected with $+$.

Output

For each test case, print `YES` in one line if $S(x)$ is a convergent sequence, or print `NO` in one line if not.

Example

standard input	standard output
2 1sinx+0cosx+3x+6/sinx 0	NO YES

Problem B. Kanade Loves Maze Designing

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Kanade is designing a mini-game. It's a puzzle game that orders players to get out of a maze. Players should also collect as many kinds of elements as they can to gain a better score.

For the easy mode, the maze can be described as a tree. There are n crossings and $n-1$ undirected passages which make the n crossings connected. The n crossings is numbered with integers from 1 to n. Exactly one element is placed on each crossing. The kind of element placed at crossing i is denoted by an integer c_i in the range $[1, n]$.

To evaluate the maze's difficulty, Kanade wants to know how many kinds of elements appear on $p(u, v)$ for every two integers $u, v \in [1, n]$. $p(u, v)$ indicates the simple path from crossing u to crossing v in the maze.

Input

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

For each test case, the first line contains one integer n ($2 \le n \le 2000$), indicating the number of crossings in the maze.

The second line contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($i < p_i$), indicating that the i-th crossing is connected with the p_i-th crossing by a passage.

The third line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le n$), indicating that the kind of element placed at crossing i is c_i.

It is promised that for all test cases, $\sum n \le 5000$.

Output

For each test case, output n lines. Each line contains two integers. Let $a_{i,j}$ be the number of kinds of elements appear on $p(i, j)$. Let

$$f(i, x) = \sum_{j=1}^{n} a_{i,j} x^{j-1}$$

Then for the i-th line, output $f(i, 19560929) \bmod (10^9 + 7)$ and $f(i, 19560929) \bmod (10^9 + 9)$, space separated.

Example

standard input	standard output
1 6 1 1 2 2 3 1 1 4 5 1 4	495644981 518101442 495644981 518101442 397599492 896634980 612255048 326063507 495644981 518101442 397599492 896634980

Note

Let $\mathbf{A} = (a_{i,j})_{n\times n}$, then for the example

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & 2 & 1 & 2 \\ 1 & 1 & 2 & 2 & 1 & 2 \\ 2 & 2 & 1 & 3 & 2 & 1 \\ 2 & 2 & 3 & 1 & 2 & 3 \\ 1 & 1 & 2 & 2 & 1 & 2 \\ 2 & 2 & 1 & 3 & 2 & 1 \end{bmatrix}$$

Problem C. Cycle Binary

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	512 megabytes

We can rewrite a binary string (i.e. strings where only 0 or 1 is included) s as $kp + p'$. p' is a prefix of p (p' could be empty). $+p'$ means concatenating p' to the last of kp. kp means concatenating k copies of p, where p is the unit of s.

The unit of a binary string s is when ignoring p', the non-empty string p which makes k maximal.

We define $v(s)$ as the value of s, which is k in the previous statement. For example, $v(01001001) = 2$, because 01001001 could be written as $2(010) + (01)$, and $v(11111) = 5$ since 11111 could be written as $5(1)$ and p' is empty .

Now given n, your task is to calculate the sum of the value of all binary strings whose length is exactly n. As the answer could be very large, just output the answer modulo 998244353.

Input

The first line contains a single positive integer T ($1 \le T \le 100$), indicating that there are T test cases.

Each test case contains a single positive integer n ($1 \le n \le 10^9$) in one line.

It is guaranteed that $\sum n \le 10^{10}$.

Output

For each test case, print an integer indicating the answer modulo 998244353 in a single line.

Example

standard input	standard output
5 1 2 3 114 514	2 6 12 954037435 530871613

Note

For $n = 3$,

$$\begin{aligned} & v(000) + v(001) + v(010) + v(011) + v(100) + v(101) + v(110) + v(111) \\ = \; & 3 + 1 + 1 + 1 + 1 + 1 + 1 + 3 \\ = \; & 12 \end{aligned}$$

Problem D. Display Substring

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

When we display a string on a LED screen, there may be different energy consumptions for displaying different characters. Given a string S with length n consisting of lowercase English letters and the energy consumption of each letter. Assume that the energy consumption of a string is the sum of the energy consumptions of all its characters. Please find the k-th smallest energy consumption of all the **different** substrings of S.

Note that a substring of a string S is a contiguous subsequence of S. We say two strings S and T are different if the lengths of them are different or there exists at least one position i satisfying $S[i] \neq T[i]$.

Input

The first line of the input contains an integer T ($1 \leq T \leq 2 \times 10^3$) — the number of test cases.

The first line of each test case contains two integers n ($1 \leq n \leq 10^5$) and k ($1 \leq k \leq \frac{n(n+1)}{2}$).

The second line of each test case contains a string S with length n consisting of lowercase English letters.

The third line of each test case contains 26 integers $c_a, c_b, \ldots, c_z$ ($1 \leq c_\alpha \leq 100$) — the energy consumption of each letter.

It is guaranteed that the sum of n among all test cases does not exceed 8×10^5.

Output

For each test case, output a single integer in a separate line — the k-th smallest energy consumption, or -1 if there is no answer.

Example

standard input	standard output
2 5 5 ababc 3 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 15 ababc 3 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	4 -1 1 1 1 1 1 1 1 1 1 1 1 1

Problem E. Didn't I Say to Make My Abilities Average in the Next Life?!

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

When reincarnating in the fantasy world, Kurihara asked the Creator to grant her the ability to average. But the Creator does really badly on math, so he considers "average"as half of the sum of the maximum and minimum among values.

There're n kinds of creatures in the fantasy world, numbered from 1 to n. Each creature has an ability value. The ability value of the i-th kind of creature is a_i. The Creator has m schemas of granting ability. For the i-th schema, The Creator will choose an interval $[l, r]$ $(x \le l \le r \le y)$ from a certain interval $[x, y]$ $(1 \le x \le y \le n)$ in a uniformly random way, calculate the "average"of the ability value from the l-th creature to the r-th creature in his own definition, and grant it to Kurihara. Please note that the definition of "average"here is half the sum of the maximum and minimum among values.

Kurihara would like to know the mathematical expectation of the ability value she will be granted.

Input

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

For each test case, the first line contains two integers n, m $(1 \le n, m \le 2 \times 10^5)$, indicating the number of creatures and the number of schemas of granting ability, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$, indicating the ability value of each creature.

For the next m lines, the i $(1 \le i \le m)$-th line contains two integers x, y $(1 \le x \le y \le n)$, indicating the i-th schema.

It is promised that for all test cases, $\sum n \le 3 \times 10^5, \sum m \le 3 \times 10^5$.

Output

For each test case, output m lines. On the i-th line, output the answer to the i-th schema in the fraction form modulo $10^9 + 7$ in one line. That is, if the answer is $\frac{P}{Q}$, you should output $P \cdot Q^{-1} \bmod (10^9 + 7)$, where Q^{-1} denotes the multiplicative inverse of Q modulo $10^9 + 7$.

Example

standard input	standard output
1 6 4 1 1 4 5 1 4 1 1 4 5 1 4 1 6	1 3 750000008 809523818

Problem F. Directed Minimum Spanning Tree

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Yukikaze is studying graph theory. She is fascinated by an interesting combinatorial optimization problem, called the Directed Minimum Spanning Tree Problem.

A subgraph T of a directed graph $G = (V, E)$ is called a Directed Minimum Spanning Tree (DMST) rooted at r if for every vertex v in $V - \{r\}$, there is exactly one path in T from r to v. The weight of a DMST is the sum of the weights of its edges.

For every vertex u in the given graph, Yukikaze wants you to find the weight of the Directed Minimum Spanning Tree rooted at vertex u.

Input

The input consists of several test cases. The first line of the input contains a single integer T ($1 \leq T \leq 3000$), denoting the number of test cases.

For each test case, the first line contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 2 \times 10^5$), denoting the number of vertices and the number of edges in the graph.

Each of the next m lines contains three integers u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9$), denoting the source, the target and the weight of the i-th edge. Please note that the edges are directed.

Let S_n and S_m be the sum of n and the sum of m in the input respectively. It is guaranteed that $1 \leq S_n \leq 5 \times 10^5$ and $1 \leq S_m \leq 10^6$.

Output

For each test case, output n lines. The i-th line should contain the weight of the Directed Minimum Spanning Tree rooted at vertex i. If such DMST doesn't exist, output -1.

Example

standard input	standard output
2 5 6 1 2 5 2 3 6 3 1 7 1 4 3 4 5 4 5 1 2 6 8 1 4 1 2 5 7 4 3 4 5 6 12 6 2 9 3 5 10 3 1 23 4 2 17	18 20 19 17 16 36 -1 55 58 -1 -1

Problem G. Increasing Subsequence

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

In a sequence of integers $a_1, a_2, \ldots, a_n$, if an increasing subsequence is not a subsequence of other increasing subsequences, we call it maximal. A subsequence is a sequence we can get by erasing some (possibly zero) elements from the original sequence.

Finding or counting the longest increasing subsequence is a classic problem. Now Yukikaze wants you to count the number of maximal increasing subsequences in some permutations modulo 998244353. A permutation of length n is a sequence of numbers such that every number from 1 to n appears exactly once.

Input

The first line of the input contains a single integer T ($1 \leq T \leq 10^4$), denoting the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$), denoting the length of the permutation.

The second line of each testcase contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$), denoting the permutation. It's guaranteed that every number from 1 to n appears exactly once.

The sum of n in all test cases will not exceed 2×10^5.

Output

For each test case, output a single integer denoting the number of the maximal increasing subsequences in the given permutation modulo 998244353.

Example

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

Problem H. Lawn of the Dead

Input file:	standard input
Output file:	standard output
Memory limit:	256 megabytes

One day, a zombie came to the Lawn of the Dead, which can be seen as an $n \times m$ grid. Initially, he stood on the top-left cell, which is $(1, 1)$.

Because the brain of the zombie was broken, he didn't have a good sense of direction. He could only move down from (i, j) to $(i + 1, j)$ or right from (i, j) to $(i, j + 1)$ in one step.

There were k "lotato mines" on the grid. The i-th mine was at (x_i, y_i). In case of being destroyed, he would never step into a cell containing a "lotato mine".

So how many cells could he possibly reach? (Including the starting cell)

Input

The first line contains a single integer t $(1 \leq t \leq 20)$, denoting the number of test cases.

The first line of each test case contains three integers n, m, k $(2 \leq n, m, k \leq 10^5)$ — there was an $n \times m$ grid, and there were k "lotato mines"on the grid.

Each of the following k lines contains 2 integers x_i, y_i $(1 \leq x_i \leq n, 1 \leq y_i \leq m)$ — there was a "lotato mine"at (x_i, y_i). It's guaranteed that there was no "lotato mine"at $(1, 1)$ and no mines were in the same cell.

It is guaranteed that $\sum n \leq 7 \cdot 10^5, \sum m \leq 7 \cdot 10^5$.

Output

For each test case, output the number of cells he could possibly reach.

Example

standard input	standard output
1 4 4 4 1 3 3 4 3 2 4 3	10

Note

The cells the zombie might reach are (1,1), (1,2), (2,1), (2,2), (2,3), (2,4), (3,1), (3,3), (4,1), (4,2).

Problem I. License Plate Recognition

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Little Rabbit is a college student who is studying Communication Engineering. In this term, he has a course called *Digital Image Processing*. At the end of the course, Little Rabbit needs to complete a project called *License Plate Recognition*, in which Little Rabbit should program a system to recognize the characters of a license plate in an image.

A classic License Plate Recognition system has three modules:

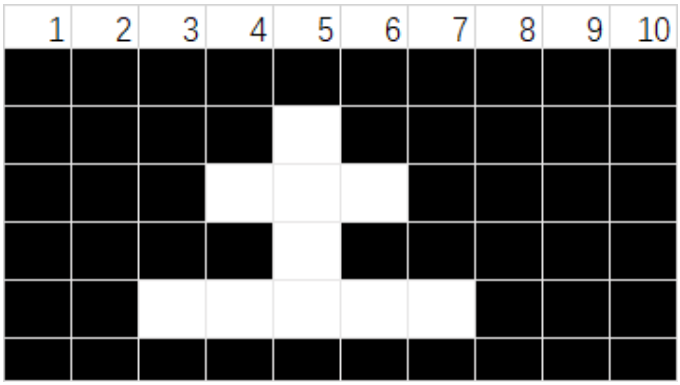
- License Plate Localization: to localize the license plate in the image.
- Character Segmentation: to segment the image so that the characters of the license plate can be separated.
- Character Recognition: to recognize the characters in the image and get the result string.

To complete the project, Little Rabbit builds a team of three members. Each member is in charge of one module. Here, Little Rabbit is in charge of the second module — Character Segmentation.

After the License Plate Localization module and some preprocessing, Little Rabbit gets some binary images that represent the license plate. The binary images are all 100 pixels in width and 30 pixels in height, containing only black pixels and white pixels. The black pixels represent the background, and the white pixels represent the characters.

Little Rabbit's task is to segment the binary images so that the characters in the images can be separated. According to the standard, there are seven characters in a license plate, lining up from left to right. Specifically, Little Rabbit's task is to find the left boundary and the right boundary of each character.

Let's consider the images as matrices with 30 rows and 100 columns. Then number the columns 1 to 100 from left to right. We define the left boundary of a character as the index of the column where the leftmost pixel is located, and the right boundary as the index of the column where the rightmost pixel is located. For example, in the following picture, the left boundary of the character is 3, and the right boundary is 7.



Now given the binary images that Little Rabbit needs to segment, please output the left boundary and the right boundary of each character. In this problem, we use `.` to represent a black pixel, and `#` to represent a white pixel.

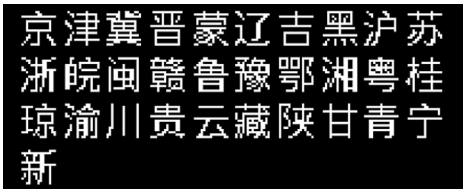
Input

The first line contains an integer T ($1 \le T \le 50$) — the number of test cases.

Each test case represents a binary image, which contains 30 lines of strings. Each line contains 100 characters, either `.` (a black pixel) or `#` (a white pixel).

Here are all the characters that may appear in the image.

Chinese characters:



ASCII version: https://paste.ubuntu.com/p/B5pTWv7s6J/

(Backup: https://github.com/cjj490168650/plate/blob/main/chn.txt)

English and numeric characters:

ASCII version: https://paste.ubuntu.com/p/59bjvwY3Yr/

(Backup: https://github.com/cjj490168650/plate/blob/main/eng.txt)

It is guaranteed that:

1. The characters in the image follow the standard of license plates. There are seven characters in the image, lining up from left to right. The first character is a Chinese character. The second character is an English character. The last five characters are English or numeric characters.
2. All characters in the image are identical to the characters given above (ASCII version), including the size and the shape.
3. There are no redundant white pixels in the image.
4. There is a spacing between characters.
5. The characters won't touch or get out of the image boundaries.

Output

For the x-th test case, output `Case #x:` in the first line.

Then in the next seven lines, the i-th line contains two integers separated by a space character, indicating the left boundary and the right boundary of the i-th character.

Note

Sample input and output: https://paste.ubuntu.com/p/28rYFsbKHb/

(Backup: https://github.com/cjj490168650/plate/blob/main/sample.txt)

Problem J. Pony Running

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Ponyville has been controlled by the changelings! The ponies are too scared to move. Fortunately, there is still a magic area in Ponyville to protect the ponies.

The magic area is an $n \times m$ grid. We denote the grid in the x-th row and y-th column by (x, y). For each second, the ponies can move up, down, left, or right with different possibilities. Formally, if a pony locates at (x, y), then in the next second, he will move up to $(x-1, y)$ with the probability of p_{xy0}, or move down to $(x+1, y)$ with the probability of p_{xy1}, or move left to $(x, y-1)$ with the probability of p_{xy2}, or move right to $(x, y+1)$ with the probability of p_{xy3}. It's possible for the ponies to move out of the magic area (for example, a pony locates at a grid in the first row, and he moves up in the next second). Since it's very dangerous to move out of the magic area, the ponies would like to know what's the sum of the mathematical expectation of the time for each pony to move out of the area.

There are q events. The first kind of event is to change the probabilities for the four directions in a grid. The second kind of event is to place a pony in each grid, then query the sum of the mathematical expectation of the time for each pony to move out of the area.

Input

The first line of input contains three integers n, m, q ($1 \le n, m, q \le 400, 1 \le n \times m \le 400$), indicating the number of rows and columns of the grid, and the number of events.

For the next $n \times m$ lines, the $((i-1) \times m + j)$-th line contains 4 integers p_{ij0}, $p_{ij1}, p_{ij2}, p_{ij3}$ ($0 \le p_{ij0}, p_{ij1}, p_{ij2}, p_{ij3} < 1,000,000,007$), indicating the probabilities for the pony that locates at (i, j) to move up, down, left, and right. The probabilities are given in the form of modulo $1,000,000,007$. That is, if the probability is P/Q, then the given integer is $P \cdot Q^{-1} \pmod{1,000,000,007}$, where Q^{-1} denotes the multiplicative inverse of Q modulo $1,000,000,007$. It's guaranteed that $p_{ij0} + p_{ij1} + p_{ij2} + p_{ij3} \equiv 1 \pmod{1,000,000,007}$.

For the next m lines, each line represents an event.

- $1\ x\ y\ p_0\ p_1\ p_2\ p_3$, indicating that the four probabilities of the gird (x, y) will be changed to $p_0\ p_1\ p_2\ p_3$. ($1 \le x \le n, 1 \le y \le m, 0 \le p_0, p_1, p_2, p_3 < 1,000,000,007, p_0 + p_1 + p_2 + p_3 \equiv 1 \pmod{1,000,000,007}$)
- 2, indicating to query the sum of the mathematical expectation of the time for each pony to move out of the area.

Output

For each query, output the answer modulo $1,000,000,007$ in a single line. That is, if the answer is P/Q, you should output $P \cdot Q^{-1} \pmod{1,000,000,007}$, where Q^{-1} denotes the multiplicative inverse of Q modulo $1,000,000,007$.

Example

standard input	standard output
2 2 5 500000004 0 500000004 0 500000004 0 500000004 0 500000004 0 500000004 0 500000004 0 500000004 0 2 1 1 1 0 500000004 0 500000004 2 1 1 2 0 500000004 0 500000004 2	500000010 14 14

Problem K. Travel on Tree

Input file:	standard input
Output file:	standard output
Memory limit:	256 megabytes

In the world of *The Three-Body Problem*, about 200 years later, people will live in huge underground tree buildings. In this problem, we use the tree data structure to describe tree buildings.

There is a tree with n nodes and $n-1$ edges with length, and n of your friends live in the tree, one at each node. You are planning to visit your friends in the next m days. Each day you choose an interval $[l, r]$ and plan to visit friends living in the nodes numbered from l to r. You can choose an arbitrary node u to start the day's visit, then travel on the tree along the edges, and finally go back to u. During the travel, you should visit all your friends living in the nodes numbered from l to r. You can visit these friends **in any order** and you can pass a node without visiting the friend. Please calculate the minimum total distance of the travel for each day.

Input

The first line of the input contains an integer T ($1 \le T \le 2 \times 10^4$) — the number of test cases.

The first line of each test case contains two integers n, m ($1 \le n, m \le 10^5$) — the number of nodes and the number of days.

Each of the following $n-1$ lines of each test case contains three integers u, v, w ($1 \le u, v \le n, 1 \le w \le 10^4$) — an edge connecting nodes u and v with length w.

Each of the following m lines of each test case contains two integers l, r ($1 \le l \le r \le n$) — a plan to visit your friends living in the nodes numbered from l to r.

It is guaranteed that for all test cases, $\sum n \le 10^6$, $\sum m \le 10^6$.

Output

For each day's plan, output the answer in a single line.

Example

standard input	standard output
2 3 3 1 2 1 2 3 1 1 1 1 2 1 3 5 5 1 2 1 1 3 2 3 4 3 3 5 4 1 3 1 4 1 5 2 5 3 5	0 2 4 6 12 20 20 14