The 2021 China Collegiate Programming Contest
Harbin Onsite

Hosted by

Harbin Engineering University

November 28, 2021

Problem A. So Many Lucky Strings

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

The first man was called George. He likes to play string concatenating games. He has a sequence $\{s_1, s_2, ...s_n\}$ which is composed of n strings. Sometimes he selects some strings from the sequence and concatenate them together in their original relative order to form a string. It means that he can arbitrarily choose a sequence of integer $A = \{a_1, a_2, .., a_k\}$, $(1 \le k \le n,\ 1 \le a_1 < a_2 < ... < a_k \le n)$, and then he will concatenate $s_{a_1}, s_{a_2}, \cdots, s_{a_k}$ together to form a new string $s_{a_1} + s_{a_2} + ... + s_{a_k}$, where $S + T$ means to put string T right after string S. For example, if his string sequence is $\{heh, e, j, ie\}$ and he chooses $\{1, 3, 4\}$,then he will get a string "hehjie".

The second man is called Karsh. He likes lucky strings. If a string is the same as itself after reversion, Karsh calls it a lucky string (or palindrome string).

Today day Karsh meets George. He hopes George could choose such a sequence A, concatenate the strings according to the chosen sequence, and finally form a lucky string for him.

Then the problem comes. How many different choices are there to form a lucky string?

Input

The first line contains an integer $n\,(1 \le n \le 100)$, denoting the number of George's strings.

Following n lines each contains a string $s_i\,(1 \le |s_i| \le 10^5)$, denoting the string sequence.

It is guaranteed that $\sum |s_i| \le 10^5$, and that all the strings only contain lower-case letters.

Output

Output one line containing an integer, denoting the number of different sequence choices to make a lucky string. Since the answer may be too large, you should output it modulo 1000000007.

Examples

standard input	standard output
4 a b aba a	8
6 hehe he he eh he jie	3

Note

For the first case, there are 8 different sequence choices to form a lucky string: $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{1, 4\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{1, 2, 3\}$.

Problem B. Magical Subsequence

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

Given a sequence $A_1, A_2, \cdots, A_n$. As for a subsequence $A_{b_1}, A_{b_2}, \cdots, A_{b_m}$ $(1 \leq b_1 < b_2 < \cdots < b_m \leq n)$, we say it magical if and only if m is even and $A_{b_1} + A_{b_2} = A_{b_3} + A_{b_4} = \cdots = A_{b_{m-1}} + A_{b_m}$. Determine the maximum length among all magical subsequences of the given sequence.

Input

The first line contains one integer n $(2 \leq n \leq 10^5)$, denoting the length of given sequence.

The second line contains n integers $A_1, A_2, \cdots, A_n$ $(1 \leq A_i \leq 100)$, denoting the given sequence.

Output

Output one line containing only one integer, denoting the answer.

Example

standard input	standard output
11 3 1 4 1 5 9 2 6 5 3 5	6

Note

One possible magical subsequence of length 6 is $\{A_1 = 3, A_5 = 5, A_7 = 2, A_8 = 6, A_9 = 5, A_{10} = 3\}$. Here $3 + 5 = 2 + 6 = 5 + 3 = 8$.

Problem C. Colorful Tree

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

Given a 1-rooted tree T, where there is a desired color c_u for each leaf(vertices who have no children) u. The leaves have no color initially, and you should do several operations to make all the leaves painted by their desired colors. For each operation, you can choose a vertex x and a color c, then make all the leaves in the x-rooted subtree painted by color c. Specially, if a leaf was painted multiple times, only the latest painted color counts. Determine the minimum possible number of operations to make all the leaves painted by their desired colors.

Input

The first line contains one integer $n\,(1 \le n \le 10^5)$, denoting the size of the tree.

The second line contains $n-1$ integers $p_1, p_2, \cdots, p_{n-1}\,(1 \le p_i \le i)$, where p_i denotes the parent of vertex $i+1$.

The third line contains n integers $c_1, c_2, \cdots, c_n\,(0 \le c_i \le n)$, denoting the desired colors, where $c_i \ne 0$ iff vertex i is leaf.

Output

Output one line containing one integer, denoting the answer.

Examples

standard input	standard output
7 1 1 2 2 3 3 0 0 0 1 1 1 2	2
7 1 1 2 2 3 3 0 0 0 1 2 1 2	3

Note

For the second test case, one possible scheme is:

1. Choose $x = 1, c = 1$, the colors become $\{0, 0, 0, 1, 1, 1, 1\}$ respectively.

2. Choose $x = 7, c = 2$, the colors become $\{0, 0, 0, 1, 1, 1, 2\}$ respectively.

3. Choose $x = 5, c = 2$, the colors become $\{0, 0, 0, 1, 2, 1, 2\}$ respectively.

Problem D. Math master

Input file:	standard input
Output file:	standard output
Time limit:	4 seconds
Memory limit:	512 megabytes

Tang Keke is good at math. She knows how to simplify fractions very well. Even greater, she invented a simplifying method by herself!

The method is to choose some digits which appear in both the top and the bottom and erase them on both sides while keeping the fraction value unchanged. The digits are the same, so they can reduce each other. Sounds very logical, right?

The method may produce multiple simplified results and the one with the least top number is called the most simplified form. Keke prepared some fractions and is going to test if you can get the most simplified form.

Note that the chosen digits form a multiset, you can see the examples for more details.

Input

The first line contains an integer n $(1 \leq n \leq 10)$, denoting the number of fractions.

Each of the next n lines contains two integer p, q $(0 < p, q < 2^{63})$, denoting the fraction $\frac{p}{q}$.

Output

For each fraction, output two integers x, y in one line, indicating the most simplified form of the corresponding fraction is $\frac{x}{y}$.

Notice: if there are some leading zeros after removal, you can ignore them and output the normal number. For example, if you get $007/123$ finally, then you should output "7 123" instead of "007 123".

Example

standard input	standard output
4	1 2
163 326	2 1
326 163	1 1
1000 1000	232 16936
2232 162936	

Note

- For the first and the second case, the erased digit multisets are both $\{3, 6\}$.

- For the third case, the erased digit multiset is $\{0, 0, 0\}$.

- For the fourth case, the erased digit multiset is $\{2\}$.

Problem E. Power and Modulo

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

Given n non-negative integers $A_1, A_2, \cdots, A_n$. Determine if there is only one positive integer M that $\forall i \in \{1, 2, \cdots, n\}$, $A_i = 2^{i-1} \bmod M$. If so, print the integer M, or print "-1".

Here, $a \bmod b$ denotes the remainder of a after division by b, where $a \in [0, b)$ holds. For example, 5 mod 2 = 1, 9 mod 3 = 0, 1 mod 2 = 1.

Input

The first line contains one integer T $(1 \leq T \leq 10^5)$, denoting the number of test cases.

For each test case:

The first line contains one integer n $(1 \leq n \leq 10^5)$, denoting the number of given integers.

The second line contains n integers $A_1, A_2, \cdots, A_n$ $(0 \leq A_i \leq 10^9)$, denoting the given integers.

It is guaranteed that $\sum n \leq 10^5$ among all cases.

Output

For each test case:

Output one line containing one integer, denoting the answer.

Example

standard input	standard output
3	7
5	-1
1 2 4 1 2	-1
5	
1 2 4 8 16	
5	
1 2 4 0 1	

Note

- For the first test case, $M = 7$.

- For the second test case, M can be any integer greater than 16, so output "-1" here.

- For the third test case, no valid M exists.

Problem F. Master Spark

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

You are playing *Touhou Impershiable Night*, a bullet curtain shooting game, and are fighting against Kirisame Marisa, the boss in stage 4B, who launchs master sparks to attack players. Following is a illustration of this scene.
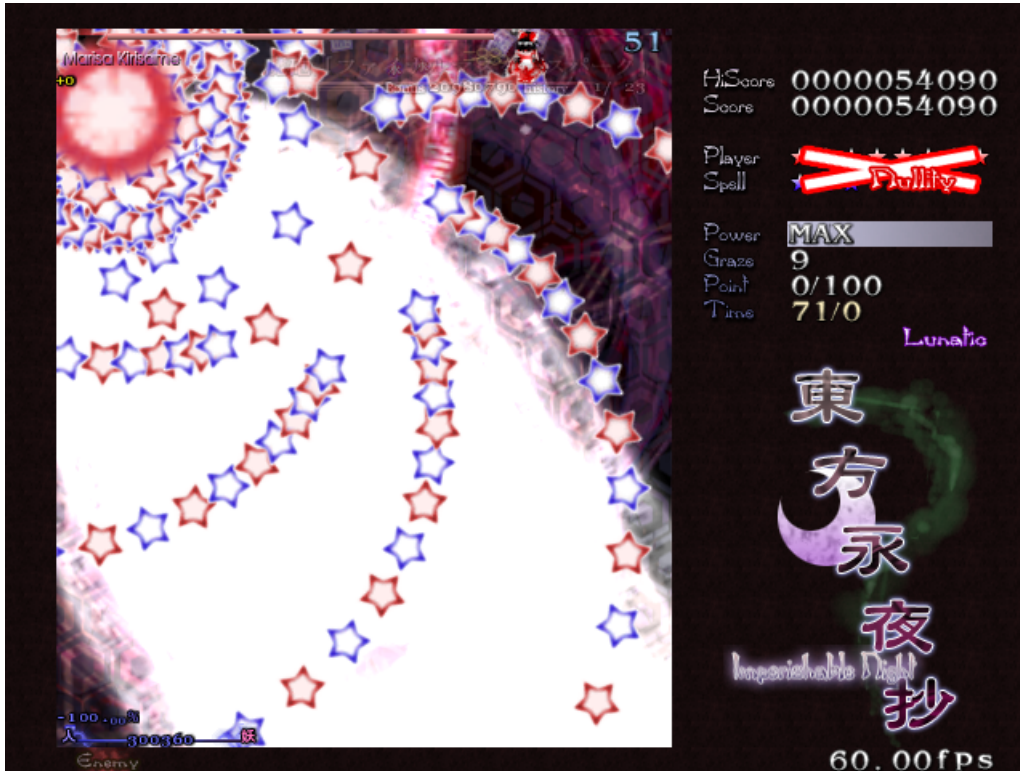

Image that the game area is in a 2D plane. The player is restricted in a rectangle region M ($\{(x, y) \mid 0 \leq x \leq X, 0 \leq y \leq Y\}$) and the sparks can be described as ribbon regions S_i ($\{(x, y) \mid C_i \leq A_i x + B_i y \leq D_i\}$).

Now given X, Y and n sparks $S_1, S_2, \cdots, S_n$, you want to know that if there exists such a point P ($P \in M$) that $\forall i \in \{1, 2, \cdots, n\}, P \notin S_i$ to avoid all sparks and autowin the game.

Input

The first line contains one integer T ($1 \leq T \leq 2000$), denoting the number of test cases.

For each test case:

The first line contatins three integers X, Y, n ($1 \leq X, Y \leq 1000, 1 \leq n \leq 2000$), denoting the size of the game region and the number of sparks.

Following n lines each contains four integers A, B, C, D ($|A|, |B|, |C|, |D| \leq 1000, A^2 + B^2 > 0, C \leq D$), denoting the parameters of given sparks.

It's guaranteed that $\sum n \leq 2000$.

Output

Output one 01-string S of length T in one line where $S_i = 1$ iff such autowinning point exists in case i while $S_i = 0$ iff no such points.
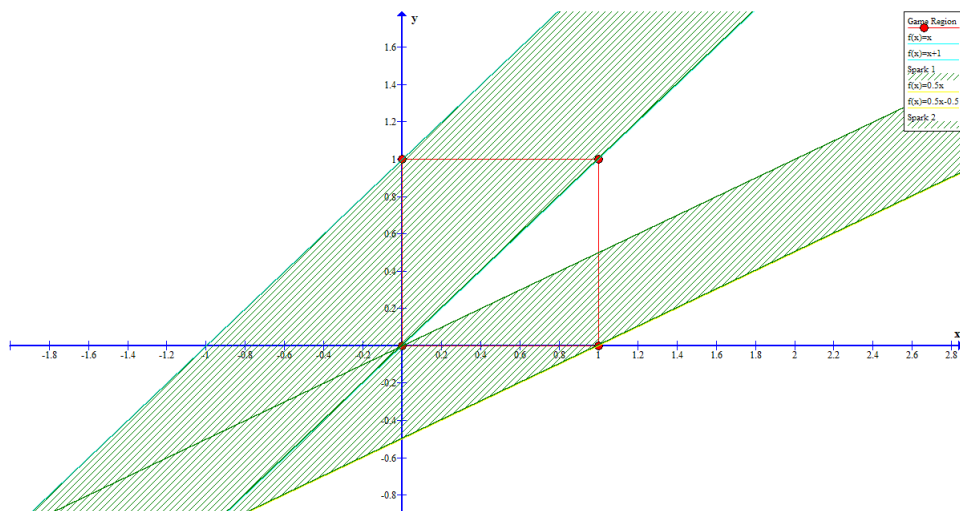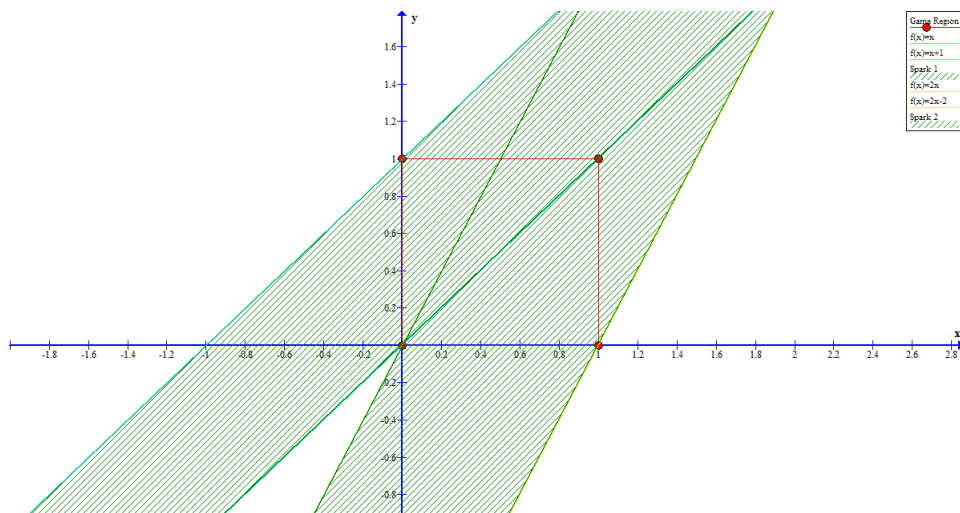
Example

standard input	standard output
2 1 1 2 1 -1 -1 0 2 -1 0 2 1 1 2 1 -1 -1 0 1 -2 0 1	01

Note

Following are the illustrations of two sample cases.

Problem G. Damaged Bicycle

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

Ring Ring Ring ... The bell rang at half past six in the morning. After turning off the alarm, George went on to sleep again. When he woke up again, it was seven fifty and there were ten minutes left for class!

George immediately got up from bed, dressed, packed his backpack, brushed his teeth and washed his face. Then, he immediately rushed out of the dormitory and embarked on the road to the teaching building. On the way, he turned on his mobile phone to locate and saw several yellow shared bicycles nearby. Therefore, he headed to a bicycle and took out his mobile phone to scan the QR code on the bicycle. Unfortunately, the bicycle wasn't unlocked, and a line of words "this bicycle is damaged and can't be unlocked" was displayed on the screen.

Without riding a bicycle, he was late. What a bad day!

Indeed, some bicycles in the school are damaged, but their location will still be displayed on the app. George always rides faster than he walks, but considering that some bicycles are damaged, if George tries one by one, it may take a long time! In this regard, he has made some modeling, and hopes you can help him find the best strategy.

The campus can be modeled as a graph of n vertices and m bidirected edges, where the i-th edge is w_i meters long. George's dormitory is located at vertex 1 and Guanghua Tower (the teaching building) is at vertex n, so George has to go from vertex 1 to vertex n to take classes. His walking speed is t meters per second and his riding speed is r meters per second. According to the bicycle sharing app, there are k parked bicycles in the campus. The i-th bicycle is located at vertex a_i, and of probability $\frac{p_i}{100}$ to be damaged according to George's experience. However, only when George arrives at vertex a_i and scans the QR code, can he determine whether the i-th bicycle is damaged or not. As soon as a bicycle is confirmed to be undamaged, George will get on it immediately and will not get off until he reaches vertex n.

Now George wants to save time to get to the classroom. So you, George's roommate, should help him find an optimal strategy to minimize the mathematical expectation of the time cost on the way, and then output this value. Or you can let him continue sleeping if vertex n is not reachable.

In this problem, you should only consider the time of walking and cycling, and you can assume that the other actions(scanning QR code, getting on, getting off, $\cdots$) cost no time.

Input

The first line contains two integers $t, r\,(1 \le t \le r \le 10^4)$ — the speed of walking and the speed of riding, respectively.

The second line contains two integers $n, m\,(1 \le n, m \le 10^5)$ — the number of vertices and the number of bidirected edges in the given graph.

Following m lines each contains three integers $u_i, v_i, w_i\,(1 \le u_i, v_i \le n, u_i \ne v_i, 1 \le w_i \le 10^4)$, denoting that vertices u, v are connected by a w_i-meter-long bidirected edge.

The next line contains a single integer $k\,(0 \le k \le 18)$, denoting the number of bicycles in campus.

Following k lines each contains two integers $a_i, p_i\,(1 \le a_i \le n, 0 \le p_i \le 100)$, denoting the locations of the bicycles and the percentages of damage probabilities respectively.

It is guaranteed that no two bicycles are in the same vertex.

Output

If George cannot reach vertex n, output one line containing one integer "-1", or output one line containing one real number, denoting the minimum expectation of the time cost on the way.

As long as the relative or absolute error between your answer and the standard answer is within 10^{-6}, your answer will be considered correct.

Examples

standard input	standard output
3 15 4 3 1 2 600 1 3 300 2 4 900 1 3 50	460.000000
3 15 5 4 1 2 600 1 3 300 2 5 900 3 4 3 2 3 50 4 0	220.600000
3 15 5 4 1 2 600 1 3 300 4 5 900 3 2 300 2 3 50 4 0	-1

Note

For the first test case, one possible strategy is:

1. Go along the route $1 \to 3$ and try to ride the only bicycle in the campus.

2. If the bicycle is damaged, go along the route $3 \to 1 \to 2 \to 4$ on foot, or go by bicycle.

Considering the time cost on the way:

- If the bicycle is damaged, George should go along the route $1 \to 3 \to 1 \to 2 \to 4$ on foot, whose total length is 2100 meters. So the time cost is $\frac{2100}{3} = 700$ seconds.

- If the bicycle is undamaged, George should go along the route $1 \to 3$ on foot, whose total length is 300 meters, and then go along the route $3 \to 1 \to 2 \to 4$ by bicycle, whose total length is 1800 meters. So the time cost is $\frac{300}{3} + \frac{1800}{15} = 220$ seconds.

As given in the input, the only bicycle has $\frac{50}{100}$ probability to be damaged. So the expectation time cost is $\frac{50}{100} \times 700 + (1 - \frac{50}{100}) \times 220 = 460$.

Problem H. What logic for?

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

What instructs and directs Reyna's analysis and cognition, is his soul together with a sort of sophisticated logic system. In the view of this system, which is blessed with awesome adaptiveness as well as adjustability, everything in a specific metaverse should be readily transformed into some kind of homogeneous expression, for instance, a string. So similarity and equivalency can be explored for further processing.

This afternoon, when wandering on a forest path in NeverLand, Reyna is absorbed by a line of peculiar silver birches. He soon realizes that some of them are essentially identical with the cue given by his logic system. To elaborate this observation, each birch is sparsely coded as a string in Reyna's mind, with a equivalency transition rule due to the sparseness.

A string S can be represented as an integer sequence $S_1, S_2, \cdots, S_l$, and according to Reyna's rule, two strings are considered k-equivalent if one can be finitely k-operated to be the other one. For each k-operation, he can choose an integer $a\,(1 \leq a \leq l - 2k + 1)$, and then swap S_{a+i} and S_{a+k+i} for all $i\,(i = 0, 1, \cdots, k - 1)$.

Now there're n query tuples $(S,\ T,\ k)$. For each tuple $(S,\ T,\ k)$, Reyna would like to query if S and T are k-equivalent.

Input

The first line contatins an integer $n\,(1 \leq n \leq 10^5)$, denoting the number of query tuples.

For each query tuple:

The first line contains $l_S + 1$ integers. The first integer $l_S\,(2 \leq l_S \leq 10^5)$ denotes the length of string S, following l_S integers $S_1, S_2, \cdots, S_{l_S}\,(1 \leq S_i \leq 10^9)$ denote the string S.

The second line contains $l_T + 1$ integers. The first integer $l_T\,(2 \leq l_T \leq 10^5)$ denotes the length of string T, following l_T integers $T_1, T_2, \cdots, T_{l_T}\,(1 \leq T_i \leq 10^9)$ denote the string T.

The third line contains one integer $k\,(1 \leq 2k \leq \min(l_S, l_T))$, denoting the transition parameter.

It's guaranteed that $\sum(l_S + l_T) \leq 10^5$.

Output

For each query tuple, output "TAK"(without quotes) in one line if S and T are equivalent, or output "NIE"(without quotes) in one line.

Example

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

Note

For the first query, one possible scheme is:

1. Choose $a = 1$, then $\{1, 2, 3, 4, 5\} \rightarrow \{3, 4, 1, 2, 5\}$

2. Choose $a = 2$, then $\{3, 4, 1, 2, 5\} \rightarrow \{3, 2, 5, 4, 1\}$

For the third query, one possible scheme is:

1. Choose $a = 1$, then $\{1, 2, 3, 4, 5\} \rightarrow \{3, 4, 1, 2, 5\}$

2. Choose $a = 2$, then $\{3, 4, 1, 2, 5\} \rightarrow \{3, 2, 5, 4, 1\}$

3. Choose $a = 1$, then $\{3, 2, 5, 4, 1\} \rightarrow \{5, 4, 3, 2, 1\}$

Problem I. Power and Zero

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

Given a sequence $A_1, A_2, \cdots, A_n$ whose elements are all positive integers. You should do some operations to make the sequence all zero. For each operation, you can appoint a sequence $B_1, B_2, \cdots, B_m$ ($B_i \in \{1, 2, \cdots, n\}$) of arbitrary length m and reduce A_{B_i} by 2^{i-1} respectively. Specially, one element in the given sequence can be reduced multiple times in one operation. Determine the minimum possible number of operations to make the given sequence all zero.

Input

The first line contains one integer T $(1 \le T \le 1000)$, denoting the number of test cases.

For each test case:

The first line contains one integer n $(1 \le n \le 10^5)$, denoting the length of given sequence.

The second line contains n integers $A_1, A_2, \cdots, A_n$ $(1 \le A_i \le 10^9)$, denoting the given sequence.

It is guaranteed that $\sum n \le 10^5$.

Output

For each test case:

Output one line containing one integer, denoting the answer.

Example

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

Note

For the first sample case, one possible scheme:

1. Appoint $B = \{1, 3, 5\}$, then A will be $\{0, 2, 1, 4, 1\}$.

2. Appoint $B = \{3, 2, 4\}$, then A will be $\{0, 0, 0, 0, 1\}$.

3. Appoint $B = \{5\}$, then A will be $\{0, 0, 0, 0, 0\}$.

For the second sample case, one possible scheme:

1. Appoint $B = \{1, 2\}$, then A will be $\{0, 2\}$.

2. Appoint $B = \{2\}$, then A will be $\{0, 1\}$.

3. Appoint $B = \{2\}$, then A will be $\{0, 0\}$.

For the third sample case, one possible scheme:

1. Appoint $B = \{1, 1, 1\}$, then A will be $\{0\}$.

Problem J. Local Minimum

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

Given a matrix of size $n \times m$. Determine the number of entries who equals the minimum value among all other entries of the same row or column. Formally, determine the value of

$$\sum_{i=1}^{n}\sum_{j=1}^{m}[\min\{M_{u,v} \mid u = i \text{ or } v = j\} = M_{i,j}]$$

Note: $[A] = 1$ if statement A is true, or $[A] = 0$.

Input

The first line contains two integers n, m $(1 \leq n, m \leq 1000)$, denoting the size of given matrix.

Following n lines each contains m integers $M_{i,j}$ $(1 \leq M_{i,j} \leq 10^6)$, denoting the given matrix.

Output

Output one line containing one integer, denoting the answer.

Example

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

Note

There are 3 entries $M_{1,1} = 1, M_{2,2} = 3, M_{3,3} = 2$ satisfying the constraint.

Problem K. Wonder Egg Priority

Input file:	standard input
Output file:	standard output
Time limit:	10 seconds
Memory limit:	1024 megabytes

Ohto Ai holds n wonder eggs. Each of them has a priority value initially. Let's say p_i is the i-th egg's priority value. The priority values will change since they are wonder eggs that have miraculous power. Ai has found the changing pattern.

Ai is going to observe the wonder eggs in the next q moments. At each moment, one of the following events will happen:

1. Ai gets the changing coefficient k of this time, and she knows for each of the l-th egg to the r-th egg, p_i will change to k^{p_i}.

2. No priority value changing happens, and Ai calculates the sum of the l-th egg to the r-th egg's priority value.

Ai wants to check if she gets the correct sum values. So for each event of the second type, she wants you to output the sum modulo M.

Input

The first line contains three integers n, q, M $(1 \leq n, q, M \leq 10^5)$, denoting the number of wonder eggs, the number of moments, and the modulus respectively.

The second line contains n integers $p_1, p_2, \cdots, p_n$ $(1 \leq p_i \leq 10^5)$, denoting the initial priority value.

Following q lines will be in one of the following format:

1. $1\ l\ r\ k$ $(1 \leq l \leq r \leq n, 1 \leq k \leq 10^5)$, denoting an event of the first type.

2. $2\ l\ r$ $(1 \leq l \leq r \leq n)$, denoting an event of the second type.

Output

For each event of the second type, print one integer in one line, denoting the answer.

Example

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

Note

- At the beginning, the priority values are $\{1, 2, 3, 4, 5\}$.

- For the first event, the answer is $2 + 3 + 4 + 5 = 14 \equiv 4 \pmod 5$.

- After the second event, the priority values are $\{1, 1, 1, 4, 5\}$.

- For the third event, the answer is $1 + 1 + 1 + 4 = 7 \equiv 2 \pmod 5$.

- After the fourth event, the priority values are $\{1, 2, 2, 16, 5\}$.

- For the fifth event, the answer is $1 + 2 + 2 + 16 + 5 = 26 \equiv 1 \pmod 5$.

- After the sixth event, the priority values are $\{1, 2, 4, 65536, 32\}$.

- For the seventh event, the answer is $1 + 2 + 4 + 65536 + 32 = 65575 \equiv 0 \pmod 5$.

Problem L. Karshilov's Matching Problem

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

This season, Karshilov was forced to retirec since he was abandoned by his teammates. However, it does not affect his enthusiasm for studying matching problems.

There are n lucky digital strings $t_1, t_2, \cdots, t_n$, where the i-th string has a lucky value w_i.

For any string S, define that $f(S) = \sum_{i=1}^{n} occur(S, t_i) \cdot w_i \pmod{998244353}$, where $occur(S, t_i)$ is the number of occurrences of string t_i in S. For example, $occur(12121, 121) = 2$ and $occur(000, 0) = 3$.

Now, Karshilov has a string S and he will do m operations one by one. Each operation is of one of the following two types:

1. 1 l c. Change all the last l characters of S, that is, $S_{|S|-l+1}, S_{|S|-l+2}, \cdots, S_{|S|}$ into c.

2. 2 l. Determine the value of $f(S[1, l])$, where $S[1, l]$ means the string formed by first l characters of S, which is $S_1 S_2 \cdots S_l$.

Can you help Karshilov to find the answers to all operations of the second type?

Input

The first line contains one integer n $(1 \leq n \leq 10^5)$, denoting the number of lucky strings.

Next n lines each contains one string t_i $(1 \leq |t_i| \leq 10^5)$ and one integer w_i $(0 \leq w_i < 998244353)$, denoting the given lucky strings and the lucky values.

The next line contains one string S $(1 \leq |S| \leq 3 \times 10^5)$, denoting the string given by Karshilov.

The next line contains one integer m $(1 \leq m \leq 3 \times 10^5)$, denoting the number of operations.

Next m lines each is of one of following two types:

1. 1 l c $(1 \leq l \leq |S|, c \in \{0, 1, \cdots 9\})$

2. 2 l $(1 \leq l \leq |S|)$

It is guaranteed that $\sum |t_i| \leq 10^5$ and the given strings only contain digitals.

Output

For each operation of the second type, output one line containing one integer, denoting the answer.

Example

standard input	standard output
2	3
479 1	6
666 2	0
666479	
5	
2 6	
1 3 6	
2 5	
1 4 2	
2 4	