Problem A. Little Q and Big Integers

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

Little Q likes positive big integers in base k notation, but not all of them. He doesn't like integers with zeroes, including leading zeroes. Additionally, he is particular about the number of occurrences of each digit. Formally, his preferences can be described as a binary matrix $g_{1..k-1,0..n}$, where for every digit i from 1 to $k-1$, if $g_{i,j} = 0$, he doesn't like integers which contain exactly j copies of digit i. He also can't accept any digit appearing more than n times. The integer must contain at least one digit.

Little Q's taste changes every day. There are m days in total, and on day i, the value g_{u_i,v_i} is flipped (0 becomes 1 and 1 becomes 0). Let $\text{cnt}(i)$ denote the number of big integers which Little Q likes after i-th day's change, and $\text{cnt}(0)$ denote the answer before all changes. Your task is to calculate the following:

$$\left(\sum_{i=0}^{m} \text{cnt}(i)\right) \bmod 786\,433.$$

Input

The first line of the input contains three integers k, n and m: the base, the upper limit and the number of days ($3 \le k \le 10$, $1 \le n \le 1.4 \cdot 10^4$, $1 \le m \le 200$).

In the next $k-1$ lines, line i contains $n+1$ integers $g_{i,0}$, $g_{i,1}$, …, $g_{i,n}$ ($0 \le g_{i,j} \le 1$). Together they provide the initial matrix g.

After that follow m lines, i-th line contains two integers u_i and v_i which mean that on i-th day, the value g_{u_i,v_i} is flipped ($1 \le u_i \le k-1$, $0 \le v_i \le n$).

Output

Print a single line with a single integer: the answer to the problem.

Example

standard input	standard output

`3 2 2`
`101`
`010`
`1 1`
`1 2`
	`13`

Problem B. Classic Quotation

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

When chatting online, we can save what somebody said to form his *classic quotation*. Little Q does this, too. And what's more, he can even change the original words. Formally, assume somebody said a string S of length n. Little Q will choose a continuous substring of S (possibly empty) and remove it, then concatenate the two remaining parts, obtaining a new string S'. For example, he might remove "`not `" from the string "`I am not SB`", so that the new string S' will be "`I am SB`".

After doing lots of such things, Little Q finds out that string T occurs as a continuous substring of S' very often.

Now given strings S and T, Little Q has k queries. Each query has the following format: given L and R, Little Q will remove a substring so that the two remaining parts are $S[1..i]$ and $S[j..n]$ where the pair of integers (i, j) is chosen equiprobably among all pairs where $1 \leq i \leq L$ and $R \leq j \leq n$. Your goal is to find E, the expected number of occurrences of T in the resulting string, and print the value $E \cdot L \cdot (n - R + 1)$.

All occurrences of T must taken into account even if they overlap. The queries are independent: the string S actually does not transform into S' and is the same for all queries.

Input

The first line of the input contains three integers n, m and k denoting the length of S, the length of T and the number of queries ($1 \leq n \leq 5 \cdot 10^4$, $1 \leq m \leq 100$, $1 \leq k \leq 5 \cdot 10^4$).

The next line contains a string S consisting of n lowercase English letters. The following line contains a string T consisting of m lowercase English letters. Each of the remaining k lines contains a query consisting of two integers L and R ($1 \leq L < R \leq n$).

Output

For each query, print a single line containing a single integer: the answer to the query.

Example

standard input	standard output
8 5 4 iamnotsb iamsb 4 7 3 7 3 8 2 7	1 1 0 0

Problem C. Counting Divisors

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	512 mebibytes

In mathematics, the function $d(n)$ denotes the number of divisors of a positive integer n.

For example, $d(12) = 6$ because 1, 2, 3, 4, 6 and 12 are all divisors of 12.

In this problem, you are given l, r and k. Your task is to calculate the following:

$$\left(\sum_{i=l}^{r} d\left(i^k\right)\right) \bmod 998\,244\,353.$$

Input

The first line of the input contains an integer T ($1 \le T \le 15$) denoting the number of test cases.

Each test case is given as a line containing three integers l, r and k ($1 \le l \le r \le 10^{12}$, $r - l \le 10^6$, $1 \le k \le 10^7$).

Output

For each test case, print a single line containing an single integer: the answer to the test case.

Example

standard input	standard output
3 1 5 1 1 10 2 1 100 3	10 48 2302

Problem D. Dirt Ratio

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

In an ACM ICPC contest, the *dirt ratio* of a team is calculated in the following way. First, let us ignore all the problems the team did not accept. Assume the team accepted X problems during the contest, and made a total of Y submissions for these problems. Then the dirt ratio is measured as $\frac{X}{Y}$. If the dirt ratio of a team is too low, the team tends to get more penalty, which is usually bad for the ranking.

Little Q is a coach, and he is now staring at the submission list of his students' team. The list contains only problem IDs and does not tell which submission is wrong and which is accepted. However, he assumed that all the problems occurring in the list were eventually solved by the team during the contest.

Little Q calculated the team's dirt ratio and felt very angry because the ratio was low. He wants to have a talk with them. To make the matter look more serious, he wants to choose a continuous subsequence of the list. He will then consider only the problems and submissions occurring in the subsequence at least once, and assume that, for each of these problems, the last submission was right, while all previous ones were wrong. Then Little Q will calculate the dirt ratio based just on the chosen continuous subsequence.

Your task is to write a program to find a subsequence which will give the lowest dirt ratio.

Input

The first line of the input contains an integer n, the length of the submission list ($1 \le n \le 6 \cdot 10^4$).

The next line contains n positive integers $a_1, a_2, \ldots, a_n$: the problem IDs of each submission ($1 \le a_i \le n$).

Output

Print a single line containing a single real number: the lowest possible dirt ratio.

The answer must be printed with an absolute error of at most 10^{-4}.

Example

standard input	standard output
5 1 2 1 2 3	0.5000000000

Problem E. Lazy Running

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

In HD University, you have to be able to run around the campus 24 times in a row: otherwise, you will fail the physical education exam and get expelled from the university. According to the rules, you must keep your speed, and your total running distance should be at least K meters.

There are four checkpoints in the campus, labeled as p_1, p_2, p_3 and p_4. Every time you pass a checkpoint, you should swipe your card, and the distance between this checkpoint and the last checkpoint you passed will be added to your total distance.

The system regards the four checkpoints as a circle: from checkpoint p_i, you can only run to one of its neighbors, p_{i-1} or p_{i+1}; p_1 and p_4 are also neighbors of each other. You can run along a straight or curved line between neighboring checkpoints, but it makes no difference for the system: only the distance between checkpoints is taken into account.

Checkpoint p_2 is the nearest to the dormitory, so Little Q always starts and ends running at this checkpoint. Please write a program to help Little Q find the shortest path such that the total running distance taken into account by the system is at least K meters.

Input

The first line of the input contains five integers K, $d_{1,2}$, $d_{2,3}$, $d_{3,4}$ and $d_{4,1}$ denoting the required distance and the distances between every pair of neighboring checkpoints ($1 \le K \le 10^{18}$, $1 \le d \le 3 \cdot 10^4$).

Output

Print a single line containing a single integer: the length of the shortest path.

Example

standard input	standard output
2000 600 650 535 380	2165

Problem F. Logical Chain

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	512 mebibytes

Every time you come across a problem you've never seen before, haven't you thought of something that is familiar to you? If so, you might think of something else, then more and more things will come to your mind. This is what is called a *logical chain*. Lu Xun's work also described this interesting phenomenon.

Assume there are n concepts, labeled by 1, 2, …, n. Little Q's mind can be expressed by a $n \times n$ binary matrix g. If he can think of concept j when he comes across concept i, then $g_{i,j}$ is 1, otherwise it is 0. For two different concepts u and v, if u can lead to v directly or indirectly and v can also lead to u directly or indirectly, then the pair (u, v) is called a *looping pair*.

Little Q's mind changes all the time. On i-th day, a total of k_i positions (u, v) in matrix g are flipped (0s become 1s and 1s become 0s). Your task is to write a program to find the number of looping pairs each day after all changes on that day.

When counting, pairs (u, v) and (v, u) should be considered the same.

Input

The first line of the input contains two integers n and m, the number of concepts and the number of days ($1 \le n \le 250$, $1 \le m \le 25\,000$).

In the next n lines, line i contains n integers $g_{i,1}$, $g_{i,2}$, …, $g_{i,n}$ ($0 \le g_{i,j} \le 1$, $g_{i,i} = 0$). Together, these lines define the matrix g.

Each of the next m parts describes a single day and starts with a line with a single integer k_i, the number of changes which happened on day i ($1 \le k_i \le 10$). Each of the next k_i lines contains two integers u and v denoting a changed position in g ($1 \le u, v \le n$, $u \ne v$).

It is guaranteed that each position will be changed at most once per day.

Output

For each day, print a single line containing a single integer: the number of looping pairs after all changes of that day.

Example

standard input	standard output
4 2 0010 1000 0000 0000 3 1 4 3 2 1 2 2 4 3 2 3	3 6

Problem G. Matching In Multiplication

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

In the mathematical discipline of graph theory, a *bipartite graph* is an undirected graph whose vertices can be divided into two disjoint sets U and V such that every edge connects some vertex in U to some vertex in V. The vertex sets U and V are both independent sets, and are usually called the parts of the graph. Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles. A *matching* in a graph is a set of edges without common vertices. A *perfect matching* is a matching such that each vertex is covered by an edge from the matching.

Little Q misunderstood the definition of bipartite graph. He thinks the size of U is equal to the size of V, and for each vertex p in U, there are exactly two edges from p. Based on such weighted graph, he defines the weight of a perfect matching as the product of weights of all the edges included in the matching, and the weight of a graph as the sum of all the perfect matchings' weights.

Your task is to write a program to compute the weight of a weighted graph made by Little Q.

Input

The first line of the input contains an integer n denoting the size of U ($1 \le n \le 3 \cdot 10^5$). The vertices in U and V are labeled separately by the integers $1, 2, \ldots, n$.

In the next n lines, the i-th line contains four integers $v_{i,1}$, $w_{i,1}$, $v_{i,2}$ and $w_{i,2}$ which mean that there is an edge between U_i and $V_{v_{i,1}}$ with weight $w_{i,1}$, and there is another edge between U_i and $V_{v_{i,2}}$ with weight $w_{i,2}$ ($1 \le v_{i,j} \le n$, $1 \le w_{i,j} \le 10^9$).

It is guaranteed that the given graph has at least one perfect matching, and there is at most one edge between every pair of vertices.

Output

Print a single line containing a single integer: the weight of the given graph. Since the answer may be very large, print it modulo 998 244 353.

Example

standard input	standard output
2 2 1 1 4 1 4 2 3	16

Problem H. Phone Call

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

There are n houses in Bytetown, labeled by 1, 2, ..., n. There is one person living in each house. Little Q lives in house 1. There are $n-1$ bidirectional streets connecting these houses, forming a tree structure. In this problem, $S(u, v)$ denotes the set containing all the houses on the shortest path from house u to house v.

The Bytetown's phone line network consists of m different lines. The i-th line can be described by five integers a_i, b_i, c_i, d_i and w_i which mean that, for any two different houses u and v from the set $S(a_i, b_i) \cup S(c_i, d_i)$, u and v can have a phone call which costs w_i dollars.

Little Q is now planning to hold a big party in his house, and he wants to invite as many people as possible. Everyone who knows about the invitation can make any number of phone calls to others to spread the invitation, but nobody can leave their house.

Write a program to find the maximum number of people that can join the party, and the minimum total cost to reach this maximum number of people. Little Q should be counted in the answer.

Input

The first line of the input contains two integers n and m: the number of houses and the number of phone lines ($1 \leq n, m \leq 10^5$).

Each of the next $n-1$ lines contains two integers u and v, denoting a bidirectional street between houses u and v. It is guaranteed that the houses and the streets form a tree.

In the next m lines, the i-th line contains five integers a_i, b_i, c_i, d_i and w_i describing a phone line ($1 \leq a_i, b_i, c_i, d_i \leq n$, $1 \leq w_i \leq 10^9$).

Output

Print a single line containing two integers: the maximum number of people that can join the party and the minimum total cost to reach that maximum number.

Example

standard input	standard output
5 2 1 2 1 3 2 4 2 5 1 3 2 4 100 2 2 4 2 10	4 210

Note

One possible solution is as follows.

Step 1: house 1 makes a phone call to house 2 using line 1, the cost is 100.

Step 2: house 1 makes a phone call to house 3 using line 1, the cost is 100.

Step 3: house 2 makes a phone call to house 4 using line 2, the cost is 10.

Problem I. Questionnaire

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

In order to get better results in official ACM ICPC contests, the team leader came up with a questionnaire. He asked every participant whether they want to have more training.

Obviously, many people don't want more training, so the clever leader didn't write down their words such as "Yes" or "No". Instead, he let everyone choose a positive integer a_i to represent their opinion. Amazingly, all the resulting numbers were distinct.

Now the leader wants to choose a pair of positive interges m ($1 < m \leq 10^9$) and k ($0 \leq k < m$), and regard those people whose number is exactly k modulo m as "Yes", and all others as "No". If there are at least as many "Yes" answers as "No" answers, the leader can have a chance to offer more training.

Please help the team leader to find such pair of m and k.

Input

The first line of the input contains an integer n: the number of ACM ICPC participants ($3 \leq n \leq 10^5$).

The next line contains n distinct integers a_1, a_2, $\ldots$, a_n: the numbers chosen by the participants ($1 \leq a_i \leq 10^9$).

Output

Print a single line containing two integers m and k. If there are several possible solutions, print any one of them.

Example

standard input	standard output
6 23 3 18 8 13 9	5 3

Problem J. Security Check

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

In the airport of Bytetown, there are two long queues waiting for the security check. Checking a person takes one minute, and the two queues can be checked at the same time.

Two teams A and B are going to travel by plane. Each team has n players, ranked from 1 to n according to their average performance. No two players in the same team share the same rank. Team A is waiting in queue 1 while team B is waiting in queue 2. Nobody else is waiting for the security check.

Little Q is the policeman who manages two queues. Every minute, he can either check the first person from one of the queues, or check the first persons from both queues at the same time. He can't change the order in the queues because that will make people unhappy. There is an additional complication, however: if two players A_i and B_j are being checked at the same time, and their ranks are almost the same, specifically $|A_i - B_j| \leq k$, they will make a lot of noise. Little Q should never let that happen.

Please write a program to help Little Q find a way to check all the people so that the required time is minimum possible.

Input

The first line of the input contains two integers n and k: the number of players in each team and the rank similarity parameter ($1 \leq n \leq 6 \cdot 10^4$, $1 \leq k \leq 10$).

The second line contains n distinct integers $A_1, A_2, \ldots, A_n$: the first queue from front to rear ($1 \leq A_i \leq n$).

The third line contains n distinct integers $B_1, B_2, \ldots, B_n$: the second queue from front to rear ($1 \leq B_i \leq n$).

Output

Print a single line containing a single integer: the minimum time required to check all the people.

Example

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

Note

One possible solution is as follows.

Minute 1: check A_1.

Minute 2: check A_2.

Minute 3: check A_3.

Minute 4: check A_4 and B_1.

Minute 5: check B_2.

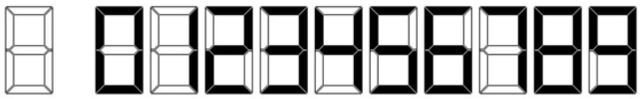
Minute 6: check B_3.

Minute 7: check B_4.

Problem K. Time to get up!

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

Little Q's alarm is ringing! It's time to get up now! However, after reading the time on the clock, Little Q lies down and starts sleeping again. Well, he has five alarms, and it's just the first one, he can continue sleeping for a while.



Little Q's clock uses a standard 7-segment LCD display for all digits, plus two small segments for the "`:`", and shows all times in 24-hour format. The "`:`" segments are on at all times.

Your job is to help Little Q read the time shown on his clock.

Input

The first line of the input contains an integer T: the number of test cases ($1 \le T \le 1440$).

Each test case is given on seven lines as a 7×21 ASCII image of the clock screen.

Each digit segment is represented by two characters, and each colon segment is represented by one character. Character "`X`" indicates a lit segment, character "`.`" indicates a dark segment or empty space. See the sample input for details.

Output

For each test case, print a single line containing a string t formatted as "`HH:MM`": the time shown on the clock (`00:00` $\le t \le$ `23:59`).

Example

standard input	standard output

`1`
`.XX...XX.....XX...XX.`
`X..X....X......X.X..X`
`X..X....X.X....X.X..X`
`......XX.....XX...XX.`
`X..X.X....X....X.X..X`
`X..X.X.........X.X..X`
`.XX...XX.....XX...XX.`
	`02:38`

Problem L. Wavel Sequence

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

Have you ever seen a wave? It's a wonderful display of nature. Little Q is attracted to this wonderful thing, he even likes everything that looks like a wave. Formally, he says that a sequence $a_1, a_2, \ldots, a_n$ is a *wavel* if and only if $a_1 < a_2 > a_3 < a_4 > a_5 < a_6 \ldots$.

Now, given two sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_m$, Little Q wants to find two sequences $f_1, f_2, \ldots, f_k$ and $g_1, g_2, \ldots, g_k$ ($1 \le f_i \le n$, $f_i < f_{i+1}$ and $1 \le g_i \le m$, $g_i < g_{i+1}$) such that $a_{f_i} = b_{g_i}$ always holds and the sequence $a_{f_1}, a_{f_2}, \ldots, a_{f_k}$ is a wavel. Moreover, Little Q is wondering how many pairs of such sequences f and g exist. Please write a program to help him figure out the answer.

Input

The first line of the input contains two integers n and m: the lengths of a and b, respectively ($1 \le n, m \le 2000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$: the sequence a ($1 \le a_i \le 2000$).

The third line contains m integers $b_1, b_2, \ldots, b_m$: the sequence b ($1 \le b_i \le 2000$).

Output

Print a single line containing a single integer: the answer to the problem. As the answer can be very large, print it modulo 998 244 353.

Example

standard input	standard output
3 5 1 5 3 4 1 1 5 3	10

Notes

Here is the list of such sequences.

(1) $f = (1)$, $g = (2)$.

(2) $f = (1)$, $g = (3)$.

(3) $f = (2)$, $g = (4)$.

(4) $f = (3)$, $g = (5)$.

(5) $f = (1, 2)$, $g = (2, 4)$.

(6) $f = (1, 2)$, $g = (3, 4)$.

(7) $f = (1, 3)$, $g = (2, 5)$.

(8) $f = (1, 3)$, $g = (3, 5)$.

(9) $f = (1, 2, 3)$, $g = (2, 4, 5)$.

(10) $f = (1, 2, 3)$, $g = (3, 4, 5)$.

Problem M. Yuno And Claris

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	512 mebibytes

Yuno failed in a contest, so she was forced to wear a JK dress. Claris won the contest, so she bought some JK dresses for Yuno to wear. Each dress has a price. Because Claris has lots of money, she bought n dresses and put them in an array $a_1, a_2, \ldots, a_n$.

Because Yuno loves data structures, she invented two kinds of operations:

- "`1 l r x y`": For all the dresses in $a_l, a_{l+1}, \ldots, a_r$, if the price of a dress is x, change its price to y.
- "`2 l r k`": Yuno wants to wear the k-th cheapest dress from $a_l, a_{l+1}, \ldots, a_r$, so tell her the price of this dress.

Input

The first line of the input contains two integers n and m: the number of dresses and the number of operations ($1 \le n, m \le 10^5$). The second line contains n integers $a_1, a_2, \ldots, a_n$: the prices of the dresses ($1 \le a_i \le n$). Each of the following m lines describes an operation. If it is a modification, then the line is formatted as "`1 l r x y`", where $1 \le l \le r \le n$ and $1 \le x, y \le n$. If it is a query, then the line is formatted as "`2 l r k`", where $1 \le l \le r \le n$ and $1 \le k \le r - l + 1$.

Output

For each query, print a single line with a single integer: the answer to the query.

Example

standard input	standard output
3 3 2 3 3 2 1 3 1 1 1 3 3 1 2 1 3 2	2 1