Problem A. Acridity and Sourness

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

You are working in the development department of Incredible Condiment Product Corporation. This company currently sells n kinds of condiments numbered 1 through n. The condiment i has acridity a_i and sourness s_i.

A recent market research revealed that consumers desire a new condiment of acridity x and sourness y, though none of the n condiments has such a taste. Here, you wonder whether such a condiment can be manufactured by mixing two different condiments. If two condiments are mixed to create a new one, its acridity and sourness are the weighted means of the two. More precisely, by mixing p grams of condiment c and q grams of condiment d, where p and q are any positive real numbers, the acridity and sourness of the new condiment become $\frac{pa_c+qa_d}{p+q}$ and $\frac{ps_c+qs_d}{p+q}$, respectively.

Please find all the possible unordered pairs of condiments such that, by mixing those two in some ratio, you can create a condiment of acridity x and sourness y.

Input

The first line of the input contains an integer n ($2 \leq n \leq 50$) representing the number of condiments that your company currently sells.

Each of the following n lines contains two integers a_i and s_i ($0 \leq a_i, s_i \leq 50$) representing the acridity and sourness of condiment i.

The last line contains two integers x and y ($0 \leq x, y \leq 50$) representing the acridity and sourness of the condiment that consumers desire.

It is guaranteed that $(a_i, s_i) \neq (x, y)$ for any i ($1 \leq i \leq n$).

Output

Print the answer in the following format.

In the first line, print m, the number of all pairs of condiments such that, by mixing those two in some ratio, you can create a condiment of acridity x and sourness y.

Then print m pairs as two integers c_i and d_i ($1 \leq c_i < d_i \leq n$) denoting the numbers of condiments in each pair.

The pairs must be printed in lexicographical order. More precisely, for any i and j ($1 \leq i < j \leq m$), either of the following properties must hold.

- $c_i < c_j$
- $c_i = c_j$ and $d_i < d_j$

Examples

standard input	standard output
8 8 6 4 8 6 0 10 5 3 7 6 50 7 7 8 6 6 7	5 1 2 2 4 2 8 3 6 5 7
6 10 20 10 30 20 10 30 10 0 0 49 50 10 10	0

Problem B. Broken Key

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You have a keyboard with 25 keys. Initially, key i ($1 \le i \le 25$) is mapped to the i-th lowercase English letter, i.e., key 1 to '`a`', key 2 to '`b`', ..., and key 25 to '`y`'. Note that initially no key is mapped to the letter '`z`'. You also have an empty string t.

You can perform the following two operations any number of times, in any order:

1. Choose an integer i ($1 \le i \le 25$) and a lowercase English letter c, and change the mapping of key i to c. This operation costs 1.
2. Choose an integer i ($1 \le i \le 25$), and append the letter currently mapped to key i to the end of t. This operation costs 0.

You are given a string s consisting of lowercase English letters. Find the minimum total cost required to make t equal to s.

Input

The only line of the input contains a string s consisting of lowercase English letters.

The length of s is between 1 and 500 000, inclusive.

Output

Print the minimum total cost as an integer.

Examples

standard input	standard output
`meatthezoo`	1
`zxcvbnmqwertyuiopasdfghjklzxcvbnm`	2

Problem C. Counting Triangles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given n distinct points on a two-dimensional plane. The i-th point is located at (x_i, y_i).

For each integer $k \geq 1$, let $f(k)$ be the maximum number of non-degenerate triangles you can place under the following conditions:

- You add k new points on the plane such that all $n + k$ points are distinct. New points can have arbitrary real-valued coordinates.
- Each triangle has its vertices among the $n + k$ points.
- No two triangles have an intersection with a positive area.

Compute $\left(\sum_{k=1}^{K} f(k)\right) \bmod 998244353$.

Input

The first line of the input contains two integers n and K ($1 \leq n \leq 200\,000$, $1 \leq K \leq 10^9$), representing the number of points and the maximum value of k.

Each of the next n lines contains two integers x_i and y_i ($0 \leq x_i, y_i \leq 10^9$), representing the coordinates of the i-th point.

It is guaranteed that all n points are distinct.

Output

Print one integer — the answer to the problem.

Examples

standard input	standard output
5 1 0 0 0 20 20 20 20 0 10 10	6
5 20250914 0 0 0 100 20 25 9 14 50 0	894241420

Problem D. Do the Inspection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

As a strange house inspector, you are inspecting a house consisting of n rooms. Each room is a rectangle in the xy-plane, and each of its edges is parallel to the x- or y-axis. Two rooms can touch but do not overlap by a positive area.

Let us say that two rooms are adjacent if their borders share a segment of positive length. It is guaranteed that one can reach any room from any other room by repeatedly moving to an adjacent room. In addition, if the borders of two rooms share only a single point, there is another room that is adjacent to both rooms.

Figure 1 depicts Sample Input 1. On the other hand, Figures 2 and 3 are invalid inputs.

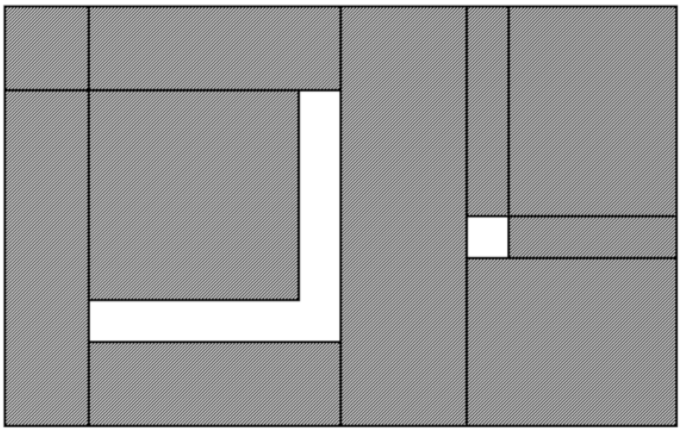


Figure 1: Illustration of Sample Input 1

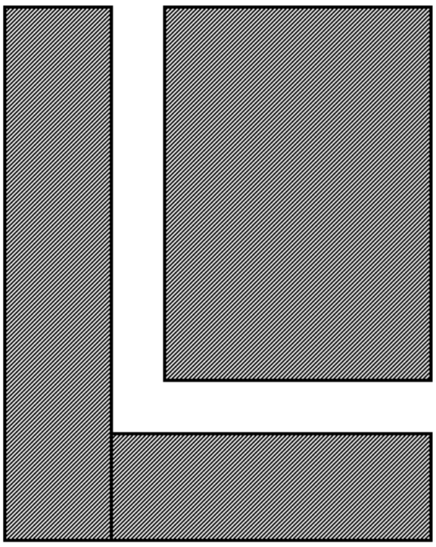


Figure 2: Illustration of an invalid input

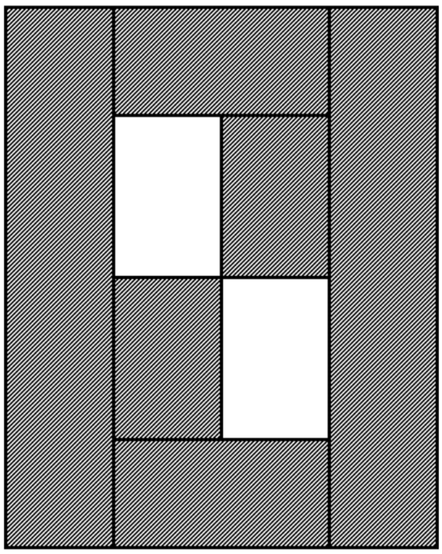


Figure 3: Illustration of another invalid input

In Figure 1, you may have found strange spaces surrounded by rooms. More precisely, a simple polygon is called a *strange space* if the following conditions are satisfied.

- The polygon and any room do not overlap by a positive area.
- For any point on the border of the polygon, there is a room whose border contains that point.

Your task is to find all the strange spaces of the house. Output the number of strange spaces and the sum of their areas.

Input

The first line of the input contains an integer n ($1 \leq n \leq 200\,000$) representing the number of rooms in the house.

Each of the following n lines contains four integers satisfying $0 \leq l_i < r_i \leq 10^9$ and $0 \leq b_i < t_i \leq 10^9$. Each line represents the corners of the i-th room: (l_i, b_i), (r_i, b_i), (r_i, t_i), and (l_i, t_i).

These rooms satisfy all the conditions explained in the problem statement.

Output

Print two numbers: the number of strange spaces in the house and the sum of their areas. It can be shown that both numbers are integers.

Examples

standard input	standard output
10 10 30 10 90 30 90 10 30 10 30 90 110 30 90 90 110 30 80 40 90 90 120 10 110 120 170 10 50 120 130 60 110 130 170 60 110 130 170 50 60	2 1200
7 0 3 0 24 3 9 0 24 9 15 0 12 9 15 12 24 15 17 0 8 15 17 8 16 15 17 16 24	0 0

Problem E. Emerald Shop

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

A shop sells n emeralds numbered $1, 2, \ldots, n$.

There are m people who visit the shop one after another. When person i arrives, they first look at which emeralds are currently for sale, and then act as follows:

- They buy emerald a_i if it is available.
- Otherwise, they buy emerald b_i if it is available.
- Otherwise, they buy nothing and leave.

Note that it is possible that $a_i = b_i$.

There are $m!$ possible arrival orders for the m people. Compute the number of arrival orders for which every person is able to buy an emerald. Output that number modulo $998\,244\,353$.

Input

The first line contains two integers n ($1 \le n \le 200\,000$), representing the number of emeralds sold in the store, and m ($1 \le m \le n$), representing the number of people visiting the store.

Each of the following m lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$).

Output

Print one integer: the answer to the problem.

Examples

standard input	standard output
4 3 2 1 3 2 3 4	4
6 6 2 3 4 3 5 4 2 5 5 1 6 6	198

Problem F. Function and String

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Consider a binary string t.

You may apply the following three operations on t any number of times (possibly zero), in any order:

- **Operation 1:** Swap two adjacent characters.
- **Operation 2:** Choose a contiguous substring "`00`" and replace it with "`1`".
- **Operation 3:** Choose a contiguous substring "`11`" and replace it with "`0`".

Let $f(t)$ be the minimum number of operations required to make the string t equal to either "`0`", "`1`", or "`01`". If it is impossible to transform t into any of these strings, we define $f(t) = 0$.

You are given a binary string $s_1 s_2 \ldots s_n$.

Compute

$$\left(\sum_{1 \le l \le r \le n} f(s_l s_{l+1} \ldots s_r) \right) \bmod 998\,244\,353.$$

Input

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

The second line contains a binary string $s_1 s_2 \ldots s_n$. Each character s_i ($1 \le i \le n$) is either '`0`' or '`1`'.

Output

Print one integer — the answer to the problem.

Examples

standard input	standard output
4 0100	10
10 1110001100	152

Problem G. Giant Playlist

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are going to go driving with m friends. To enjoy the driving, you have a playlist of n songs numbered 1 through n.

At time 0, you choose one of the songs and start it from its beginning. Then, the playlist repeats forever. For each k ($1 \le k \le n$), once the song k starts, it lasts l_k units of time, and then the song $k+1$ (or the song 1 if $k = n$) follows immediately.

The i-th friend, who loves the song f_i, joins you at time $t_i - 0.5$. After that, they get excited whenever the song f_i starts. Note that even if the song f_i is already being played when they join you, they don't get excited until their favorite song starts playing from the beginning.

If you choose the first song to play optimally, when is the earliest time that all the m friends get excited at least once? Note that you cannot start playing a song from the middle.

Input

The first line of the input contains two integers n and m ($1 \le n \le 200\,000$, $1 \le m \le 200\,000$), where n denotes the number of songs in the playlist and m denotes the number of friends.

The second line contains n positive integers $l_1, l_2, \ldots, l_n$ ($1 \le l_i \le 10^{15}$). Each l_i represents the length of song i. Their sum does not exceed 10^{15}.

The third line contains m integers $t_1, t_2, \ldots, t_m$ ($1 \le t_i \le 10^{15}$). Each t_i represents that the i-th friend joins you at time $t_i - 0.5$.

The fourth line contains m integers $f_1, f_2, \ldots, f_m$ ($1 \le f_i \le n$). Each f_i is the number of the favorite song of the i-th friend.

Output

Output the earliest time that all the m friends get excited at least once. It can be shown that the answer is an integer.

Examples

standard input	standard output
3 4 3 1 4 10 7 3 7 1 3 2 1	12
5 7 20 25 9 14 20 25 9 14 75 38 100 38 3 1 1 5 2 4 4	136

Problem H. How Many Ways to Pack?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You work at the Identifiability in Container Packing Center, where you research the uniqueness of container packings.

You are given n items. Item i has a positive integer weight a_i ($1 \le i \le n$).

You consider packing a (nonempty) subset $s \subseteq \{1, 2, \ldots, n\}$ of the items into containers. You may use any number of nonempty containers (empty containers are not allowed). Fix a positive integer w denoting the capacity of each container. A valid packing of s is an assignment of the items in s to containers that satisfies all of the following:

Cover: Every item in s is placed in exactly one container.

Capacity: In each container, the total weight of its items is at most w.

Non-mergeability: For any two distinct containers a and b, the total weight of all items contained in containers a and b is strictly greater than w (i.e., no two containers can be merged into a single container without violating capacity w).

Containers are indistinguishable and items are distinct even if some have the same weight. Two packings are considered the same if and only if they induce the same partition of s; equivalently, for any distinct $i, j \in s$, items i and j are in the same container in one packing if and only if they are in the same container in the other.

For a fixed w, call a subset s *uniquely packable* if there is exactly one valid packing of s that satisfies all conditions.

You are given an integer W. Let $f(w)$ ($w = 1, 2, \ldots, W$) be the number of uniquely packable nonempty subsets for capacity w. For each $w = 1, 2, \ldots, W$, output $f(w)$ modulo 998 244 353. In other words, for each $w = 1, 2, \ldots, W$, define

$$f(w) = \#\{s \subseteq \{1, 2, \ldots, n\} \mid s \text{ is nonempty and uniquely packable for capacity } w\}.$$

For each $w = 1, 2, \ldots, W$, print $f(w)$ modulo 998 244 353.

Input

The first line of the input contains two integers n and W ($1 \le n \le 5\,000$, $1 \le W \le 5\,000$), representing the number of items and the maximum capacity to consider, respectively.

The second line contains n positive integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le W$). Each a_i represents the weight of item i.

Output

Print W integers in a single line separated by spaces: for each $w = 1, 2, \ldots, W$, the w-th integer is $f(w)$ modulo 998 244 353 (the answer for capacity w).

Examples

standard input	standard output
4 4 1 3 2 4	1 3 7 13
3 9 9 1 4	1 1 1 3 3 3 3 3 7
2 2 2 2	0 3

Problem I. In One Row

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Ryan has a rooted tree with n vertices, where each vertex has a number 0 or 1 written on it.

Ryan's friend also has a rooted tree, where each vertex has a number 0 or 1 written on it. Let the number of vertices in this tree be m.

Ryan wants to arrange these $n + m$ vertices in a horizontal row. Here, for every vertex, there should be no ancestor of that vertex to the right of that vertex. Note that there are no constraints between the vertices in Ryan's tree and the vertices in his friend's tree.

After arranging the vertices, let x be the sequence obtained by reading the numbers written on the vertices from left to right. Find the minimum possible number of inversions in x, which is the number of pairs $i < j$ such that $x_i > x_j$.

Since Ryan has q friends, solve the above problem for each of his friends.

The numbers written on the vertices of Ryan's friends' trees are given in an encrypted form. See the bottom of the Input section for details.

Input

The first line contains an integer n ($1 \le n \le 200\,000$), representing the number of vertices in Ryan's tree.

The second line contains $n - 1$ integers. Each p_i ($2 \le i \le n$, $1 \le p_i < i$) represents that the parent of vertex i is vertex p_i. Note that vertex 1 is the root of the tree and p_1 is not given.

The third line contains n integers. Each v_i ($1 \le i \le n$, $0 \le v_i \le 1$) represents the number written on vertex i.

The fourth line contains an integer q ($1 \le q \le 100\,000$), representing the number of Ryan's friends.

For each friend k ($1 \le k \le q$), the input for their tree is given in the following format:

- The first line contains an integer m_k ($1 \le m_k \le 200\,000$), representing the number of vertices in the k-th friend's tree.
- The second line contains $m_k - 1$ integers. Each $p_{k,i}$ ($2 \le i \le m_k$, $1 \le p_{k,i} < i$) represents that the parent of vertex i is vertex $p_{k,i}$. Note that vertex 1 is the root of the tree and $p_{k,1}$ is not given.
- The third line contains m_k integers. Each $u_{k,i}$ ($1 \le i \le m_k$, $0 \le u_{k,i} \le 1$) represents the encrypted number written on vertex i of the k-th friend's tree.

Additionally, the sum of the m_k ($1 \le k \le q$) does not exceed $200\,000$.

Decrypting the Numbers on the Vertices of Ryan's Friends' Trees:

Let $x_0 = 0$, and for each friend k ($1 \le k \le q$), let x_k denote the answer for the k-th friend. The actual value $v_{k,i}$ ($1 \le k \le q$, $1 \le i \le m_k$, $0 \le v_{k,i} \le 1$) written on vertex i of the k-th friend's tree is determined as follows:

$$v_{k,i} = (\text{powmod}(x_{k-1}, i, 998\,244\,353) + u_{k,i}) \bmod 2.$$

Here, $\text{powmod}(a, b, m)$ denotes $(a^b \bmod m)$.

Output

Print q lines. The i-th line should contain a single integer, representing the minimum possible inversion number for Ryan and his i-th friend.

Example

standard input	standard output
6 1 1 2 3 3 0 1 1 0 0 0 4 4 1 2 2 1 0 1 0 6 1 1 2 3 3 1 1 0 0 1 0 1 0 15 1 2 3 2 5 6 2 2 9 10 1 12 13 12 1 1 1 0 1 1 0 0 1 0 0 1 1 0 0	9 7 4 42

Note

The decrypted values corresponding to this sample are shown below.

```
6
1 1 2 3 3
0 1 1 0 0 0
4
4
1 2 2
1 0 1 0
6
1 1 2 3 3
0 0 1 1 0 1
1

1
15
1 2 3 2 5 6 2 2 9 10 1 12 13 12
1 1 1 0 1 1 0 0 1 0 0 1 1 0 1
```

Problem J. Japanese Art

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There are $2n$ points in the xy-plane. Any two points have neither the same x-coordinate nor the same y-coordinate. Each point has a color represented by an integer between 1 and n (inclusive). For each of the n colors, there are exactly two points of that color.

Hanako, the Japanese artist, is willing to create a masterpiece by drawing n polygonal chains in the xy-plane. According to her aesthetic sense, a masterpiece must satisfy all the following conditions.

- Any two points having the same color are the endpoints of one of the polygonal chains.
- Each polygonal chain consists of exactly two line segments, each of which is parallel to the x- or y-axis.
- No two polygonal chains intersect.

Your task is to determine whether Hanako can create such a masterpiece.

Input

The first line of the input contains an integer t ($1 \le t \le 50\,000$) representing the number of test cases.

The first line of each test case contains an integer n ($1 \le n \le 1\,000$) representing the number of polygonal chains which Hanako has to draw. Each of the following $2n$ lines contains two integers y_i and c_i satisfying $1 \le y_i \le 2n$ and $1 \le c_i \le n$. Each line represents that the i-th point has the coordinate (i, y_i) and the color c_i.

It is guaranteed that $y_i \ne y_j$ if $i \ne j$. In addition, for each of the n colors, there are exactly two points of that color.

The sum of n^2 over all the test cases does not exceed 10^6.

Output

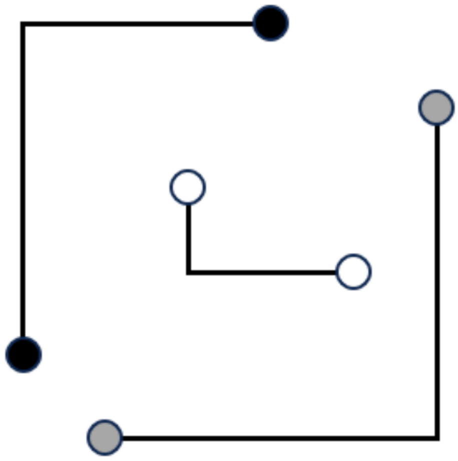
If Hanako can create a masterpiece, print "`Yes`"; otherwise, print "`No`".

Example

standard input	standard output
2 3 2 1 1 2 4 3 6 1 3 3 5 2 3 2 3 6 1 5 2 1 1 4 3 3 2	Yes No

Note

One of the possible masterpieces in Sample Input 1 is depicted below:



Problem K. Know Your Prefix

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

You have a sequence $a = (a_1, a_2, \ldots, a_n)$ of length n, where initially $a_i = i$ for all i $(1 \le i \le n)$.

There are k queries. In the q-th query, an integer x_q $(1 \le x_q \le |a|)$ is given, and you replace a with

$$(a_1, a_2, \ldots, a_{x_q}, a_1, a_2, \ldots, a_{|a|}, a_{x_q+1}, a_{x_q+2}, \ldots, a_{|a|})$$

where $|a|$ denotes the current length of a. In other words, you insert a copy of the entire sequence a right after its first x_q elements.

After processing all k queries in order, output the first m elements of the resulting sequence a.

Input

The first line of the input contains three integers n, m, and k $(1 \le n \le 10^6$, $1 \le m \le \min(10^6, n \times 2^k)$, $1 \le k \le 10^6)$.

The q-th of the following k lines contains one integer x_q $(1 \le x_q \le \min(10^{12}, n \times 2^{q-1}))$, representing the parameter of the q-th query.

Output

Print m integers $a_1, a_2, \ldots, a_m$, the first m elements of the final sequence after all queries are applied, in a single line separated by spaces.

Examples

standard input	standard output
5 9 2 2 6	1 2 1 2 3 4 1 2 1
200000 10 10 234 54 2346 374 6 24 547 65 20000000 74	1 2 3 4 5 6 1 2 3 4

Problem L. Leave Only Root

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given a tree with n vertices rooted at vertex 1, and a positive integer k. The i-th edge connects vertex a_i and vertex b_i.

We repeat the following operation until the number of vertices in the tree becomes 1 (i.e., only the root remains):

- Choose a leaf vertex (other than vertex 1) and delete it. After that, the tree changes as follows:
  - If the deleted vertex is a child of vertex 1, nothing happens.
  - Otherwise, k new copies of the subtree rooted at the parent of the deleted vertex (after the deletion) are attached to the grandparent of the deleted vertex (the parent of the parent of the deleted vertex).

Find the minimum possible number of operations to make the tree consist of only vertex 1, modulo $998,244,353$.

It can be proven that the tree can be reduced to only vertex 1 in a finite number of operations.

Input

The first line contains two integers n and k ($1 \le n \le 100\,000$, $1 \le k < 998,244,353$).

Each of the next $n-1$ lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$), representing the edges of the tree.

It is guaranteed that the edges form a tree.

Output

Print the minimum possible number of operations to make the tree consist of only vertex 1, modulo $998,244,353$.

Examples

standard input	standard output
3 2 1 2 2 3	4
5 2 1 2 1 3 2 4 2 5	14