

Problem A

Mix Condiments

Time Limit: 2 seconds

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 acidity a_i and sourness s_i .

A recent market research revealed that consumers desire a new condiment of acidity x and sourness y , though none of the n condiments has such taste. Here, you wonder whether such a condiment can be manufactured by mixing two of the condiments. If two condiments are mixed to create a new one, its acidity and sourness are the weighted means from the two. More precisely, by mixing p gram of condiment c and q gram of condiment d where p and q are any positive real numbers, the acidity 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 acidity x and sourness y .

Input

The input consists of a single test case of the following format.

```
n
a1 s1
a2 s2
⋮
an sn
x y
```

The first line 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 acidity and sourness of condiment i . The last line contains two integers x and y ($0 \leq x, y \leq 50$) representing the acidity 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.

```
m
c1 d1
c2 d2
⋮
cm dm
```

m is the number of all pairs of condiments such that by mixing those two in some ratio, you can create a condiment of acidity x and sourness y . c_i and d_i ($1 \leq c_i < d_i \leq n$) are the numbers of condiments in each pair.

The pairs must be output in the 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 $d_i < d_j$

Sample Input 1

```
8
8 6
4 8
6 0
10 5
3 7
6 50
7 7
8 6
6 7
```

Sample Output 1

```
5
1 2
2 4
2 8
3 6
5 7
```

Sample Input 2

```
6
10 20
10 30
20 10
30 10
0 0
49 50
10 10
```

Sample Output 2

```
0
```

Problem B

Broken Keyboard

Time Limit: 2 seconds

You have a keyboard with 25 keys. Initially, key i ($1 \leq i \leq 25$) is mapped to the i -th lowercase English letter, i.e., key 1 to 'a', key 2 to 'b', ..., and key 25 to 'y'. 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 ($1 \leq i \leq 25$) and a lowercase English letter c , and change the mapping of key i to c . This operation costs 1.
2. Choose an integer i ($1 \leq i \leq 25$), 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 input consists of a single test case in the following format.

S

The only line 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.

Sample Input 1

meatthezoo	1
------------	---

Sample Output 1

Sample Input 2

zxcvbnmqwertyuiopasdfghjklzxcvbnm	2
-----------------------------------	---

Sample Output 2

Problem C

Triangles

Time Limit: 2 seconds

You are given N distinct points on a 2D 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.
- 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 input consists of a single test case in the following format.

```
N K
x1 y1
⋮
xN yN
```

The first line 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 the answer.

Sample Input 1

```
5 1
0 0
0 20
20 20
20 0
10 10
```

Sample Output 1

```
6
```

Sample Input 2

```
5 20250914
0 0
0 100
20 25
9 14
50 0
```

Sample Output 2

```
894241420
```

Problem D Strange House

Time Limit: 4 seconds

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.

Let us say that two rooms are adjacent if their borders share a segment of a positive length. It is guaranteed that one can reach from any of the rooms to 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 which is adjacent to both rooms.

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

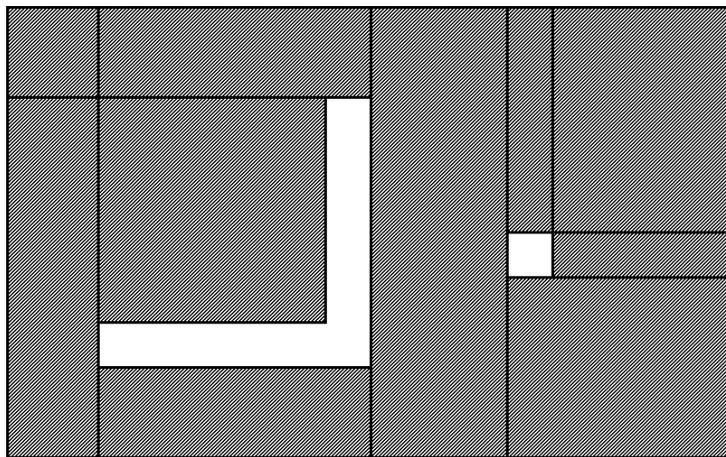


Figure D-1: Illustration of Sample Input 1

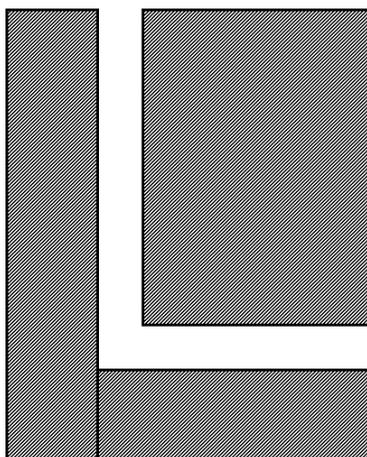


Figure D-2: Illustration of an invalid input

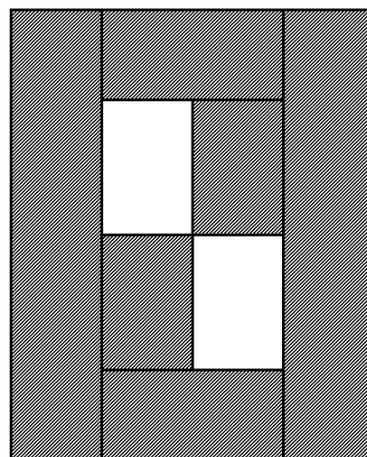


Figure D-3: Illustration of another invalid input

In Figure D-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.
- 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 input consists of a single test case of the following format.

```
n
l1 r1 b1 t1
⋮
ln rn bn tn
```

The first line 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 that corners of the i -th room are (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

Output two lines. The first line should contain the number of strange spaces of the house. The second line should contain the sum of areas of these strange spaces.

Sample Input 1

```
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
```

Sample Output 1

```
2
1200
```

Sample Input 2

```
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
```

Sample Output 2

```
0
0
```

Problem E

To All Tha Customers

Time Limit: 2 seconds

A shop sells N items numbered $1, 2, \dots, N$.

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

- They buy item A_i if it is available.
- Otherwise, they buy item 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 item. Output that number modulo 998244353.

Input

```
N M
A1 B1
A2 B2
⋮
AM BM
```

The first line contains an integer N ($1 \leq N \leq 200\,000$) representing the number of items sold in the store and M ($1 \leq M \leq N$) representing the number of people visiting the store.

Each of the following M lines contains two integers A_i and B_i ($1 \leq A_i, B_i \leq N$).

Output

Print the answer.

Sample Input 1

```
4 3
2 1
3 2
3 4
```

Sample Output 1

```
4
```

Sample Input 2

```
6 6
2 3
4 3
5 4
2 5
5 1
6 6
```

Sample Output 2

```
198
```

Problem F

00 → 1

Time Limit: 2 seconds

For any binary string T , define $f(T)$ as follows.

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 $S_1S_2 \dots S_N$.

Compute $\left(\sum_{1 \leq l \leq r \leq N} f(S_l S_{l+1} \dots S_r) \right) \bmod 998244353$.

Input

The input consists of a single test case of the following format.

N
 $S_1S_2 \dots S_N$

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

The second line contains a binary string $S_1S_2 \dots S_N$. Each character S_i ($1 \leq i \leq N$) is either ‘0’ or ‘1’.

Output

Print the answer.

Sample Input 1

4
0100

Sample Output 1

10

Sample Input 2

10
1110001100

Sample Output 2

152

Problem G

Driving Playlist

Time Limit: 2 seconds

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 ($1 \leq k \leq n$), 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 because everyone wants to enjoy their favorite song 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 input consists of a single test case of the following format.

```
n m
l1 l2 ⋯ ln
t1 t2 ⋯ tm
f1 f2 ⋯ fm
```

The first line contains two integers n and m ($1 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$). n is the number of songs in the playlist and m is the number of friends who join your driving.

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

The third line contains m integers t_1, t_2, \dots, t_m ($1 \leq t_i \leq 10^{15}$). 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, \dots, f_m ($1 \leq f_i \leq n$). Each f_i is the number of the favorite song of the i -th friend.

Output

Output an integer, which is the earliest time that all the m friends get excited at least once.

Sample Input 1

```
3 4
3 1 4
10 7 3 7
1 3 2 1
```

Sample Output 1

```
12
```

Sample Input 2

Sample Output 2

5 7 20 25 9 14 20 25 9 14 75 38 100 38 3 1 1 5 2 4 4	136
---	-----

In the first sample, if you start the playlist from song 3, the four friends first get excited at times 12, 8, 7, and 12, respectively.

Problem H

Count Unique Packing

Time Limit: 2 seconds

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 ($1 \leq i \leq N$).

You consider packing a (nonempty) subset $S \subseteq \{1, 2, \dots, N\}$ 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 weights of the items contained in A or 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 box in one packing if and only if they are in the same box 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 $f(w)$ ($w = 1, 2, \dots, W$) be the number of uniquely packable nonempty subsets for capacity w . For each $w = 1, 2, \dots, W$, output $f(w)$ modulo 998244353. In other words, for each $w = 1, 2, \dots, W$, define

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

Output W integers; for each $w = 1, 2, \dots, W$, print $f(w)$ modulo 998244353.

Input

The input consists of a single test case of the following format.

```
N W
A1 A2 ⋯ AN
```

The first line contains two integers N ($1 \leq N \leq 5000$) representing the number of items and W ($1 \leq W \leq 5000$) representing upper bound on the capacity parameter w (i.e., the maximum capacity to consider). The second line contains N positive integers A_1, A_2, \dots, A_N ($1 \leq A_i \leq W$). Each A_i represents the weight of the item i .

Output

Output W integers in a single line separated by spaces: for each $w = 1, 2, \dots, W$, the w -th integer is $f(w)$ modulo 998244353 (the answer for capacity w).

Sample Input 1

4 4 1 3 2 4	1 3 7 13
----------------	----------

Sample Output 1

Sample Input 2

3 9 9 1 4	1 1 1 3 3 3 3 7
--------------	-----------------

Sample Output 2

Sample Input 3

2 2 2 2	0 3
------------	-----

Sample Output 3

Problem I

Double 01 on Tree

Time Limit: 2 seconds

Ryan has a rooted tree with N vertices, where each vertex is written with a number 0 or 1.

Ryan's friend also has a rooted tree, where each vertex is written with a number 0 or 1. 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. Ryan wants to minimize the inversion number of X . Find the minimum possible inversion number of X .

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 input consists of a single test case of the following format.

```
N
P2 P3 ... PN
V1 V2 ... VN
Q
M1
P1,2 P1,3 ... P1,M1
U1,1 U1,2 ... U1,M1
M2
P2,2 P2,3 ... P2,M2
U2,1 U2,2 ... U2,M2
⋮
MQ
PQ,2 PQ,3 ... PQ,MQ
UQ,1 UQ,2 ... UQ,MQ
```

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

The second line contains $N - 1$ integers. Each P_i ($2 \leq i \leq N$, $1 \leq P_i < i$) 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 ($1 \leq i \leq N$, $0 \leq V_i \leq 1$) represents that the number written on vertex i .

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

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

- The first line contains an integer M_k ($1 \leq M_k$), representing the number of vertices in the k -th friend's tree.

- The second line contains $M_k - 1$ integers. Each $P_{k,i}$ ($2 \leq i \leq M_k, 1 \leq P_{k,i} < i$) 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 $U_{k,i}$ ($1 \leq i \leq M_k, 0 \leq U_{k,i} \leq 1$) represents the encrypted number written on vertex i of the k -th friend's tree.

Additionally, the sum of the M_k ($1 \leq k \leq Q$) 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 ($1 \leq k \leq Q$), let X_k denote the answer for the k -th friend. The actual value $V_{k,i}$ ($1 \leq k \leq Q, 1 \leq i \leq M_k, 0 \leq V_{k,i} \leq 1$) written on vertex i of the k -th friend's tree is determined as follows:

$$V_{k,i} = (\text{powmod}(X_{k-1}, i, 998244353) + U_{k,i}) \bmod 2.$$

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.

Sample Input	Sample Output
<pre>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</pre>	<pre>9 7 4 42</pre>

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

Hanako's Art II

Time Limit: 2 seconds

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.

An artist, Hanako, 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 input consists of multiple test cases. The first line of input contains an integer t ($1 \leq t \leq 50\,000$) representing the number of test cases. After that, t test cases follow. Each of them is given in the following format.

```
n
y1 c1
⋮
y2n c2n
```

The first line contains an integer n ($1 \leq n \leq 1\,000$) 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 $1 \leq y_i \leq 2n$ and $1 \leq c_i \leq n$. Each line represents that the i -th point has the coordinate (i, y_i) and the color c_i .

It is guaranteed that $y_i \neq y_j$ if $i \neq j$. In addition, no three points have the same 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".

Sample Input	Sample Output
2	Yes
3	No
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	

One of the possible masterpieces in Sample Input 1 is depicted in Figure J-1.

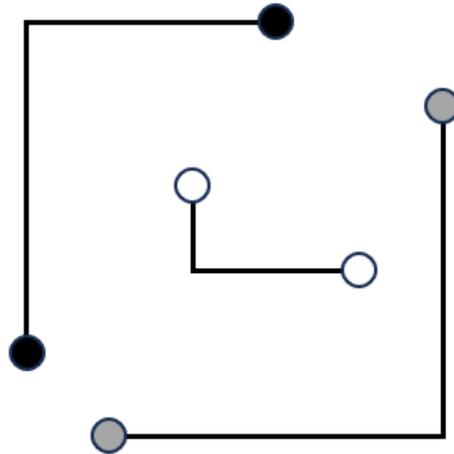


Figure J-1: Illustration of Sample Input 1

Problem K

All Copy Paste

Time Limit: 2 seconds

You have a sequence $A = (A_1, A_2, \dots, A_N)$ of length N , where initially $A_i = i$ for all i ($1 \leq i \leq N$).

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

$$(A_1, A_2, \dots, A_{x_q}, A_1, A_2, \dots, A_{|A|}, A_{x_q+1}, A_{x_q+2}, \dots, A_{|A|}),$$

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 Q queries in order, output the first M elements of the resulting sequence A .

Input

The input is given in the following format.

```
N M Q
x1
x2
⋮
xQ
```

The first line contains three integers N , M , and Q ($1 \leq N \leq 10^6$, $1 \leq M \leq \min(10^6, N \times 2^Q)$, $1 \leq Q \leq 10^6$). Each of the following Q lines contains one integer x_q ($1 \leq x_q \leq \min(10^{12}, N \times 2^{q-1})$), representing the parameter of the q -th query.

Output

Print M integers A_1, A_2, \dots, A_M , the first M elements of the final sequence after all queries are applied, in a single line separated by spaces.

Sample Input 1

```
5 9 2
2
6
```

Sample Output 1

```
1 2 1 2 3 4 1 2 1
```

Sample Input 2

```
200000 10 10
234
54
2346
374
6
24
547
65
20000000
74
```

Sample Output 2

```
1 2 3 4 5 6 1 2 3 4
```

Problem L

Arrange One More Office

Time Limit: 2 seconds

Your company offers a rental office service on a floor of a building. The floor, which forms a rectangle, is partitioned into square sections of the same size in a grid form. Two sections are said to be adjacent if they share an edge. Some sections may contain pillars. Some sections may be empty for future use. Any other sections are used as offices, where each office consists of two adjacent sections without pillars.

One day, a potential customer came to your company and applied for an office. Because all the existing offices are already rented, you need to set up a new one. Of course, if there are two adjacent empty sections, you can set up an office there. But if not, it may be possible to relocate some of the existing offices to place one more office. That being said, because your company must pay compensation to the affected renters, the rearrangement should minimize the number of affected offices.

More formally, your task is to find a rearrangement plan that satisfies the following conditions.

- The floor contains $k + 1$ offices after the rearrangement, where k is the number of offices before the rearrangement.
- The set of sections with pillars does not change.
- Each office still consists of two adjacent sections without pillars.
- No two offices overlap.
- Among all the possible rearrangement plans satisfying all the above conditions, the number of unaffected offices is maximized. Here, an office, which consists of some two sections before the rearrangement, is unaffected by the rearrangement if these two sections still belong to the same office after the rearrangement.

Please find such a plan if it exists. If there can be multiple plans, output any of them.

Input

The input consists of multiple test cases. The first line of input contains an integer t ($1 \leq t \leq 500\,000$) representing the number of test cases. After that, t test cases follow. Each of them is given in the following format.

```
h w
s1,1s1,2⋯s1,w
s2,1s2,2⋯s2,w
⋮
sh,1sh,2⋯sh,w
```

The first line contains two integers h and w ($h \geq 1$, $w \geq 1$, $h \times w \leq 500\,000$) representing that the floor consists of h rows and w columns of sections.

Each of the following h lines contains w characters. Let us denote by (r, c) the section in the c -th column of the r -th row. Each character $s_{r,c}$, which represents the information of the section (r, c) , is one of '#', '.', '^', 'v', '<', and '>'. If $s_{r,c}$ is '#', then (r, c) is a section with a pillar. If $s_{r,c}$ is '.', then (r, c) is an empty section. If there is an office consisting of (r, c) and $(r + 1, c)$, then $s_{r,c}$ is '^' and $s_{r+1,c}$ is 'v'. If there is an office consisting of (r, c) and $(r, c + 1)$, then $s_{r,c}$ is '<' and $s_{r,c+1}$ is '>'.

If $s_{r,c}$ is '^', it is guaranteed that $r \leq h - 1$ and $s_{r+1,c}$ is 'v'. Similarly, if $s_{r,c}$ is 'v', then $r \geq 2$ and $s_{r-1,c}$ is '^'. If $s_{r,c}$ is '<', then $c \leq w - 1$ and $s_{r,c+1}$ is '>'. Finally, if $s_{r,c}$ is '>', then $c \geq 2$ and $s_{r,c-1}$ is '<'.

The sum of $h \times w$ over all the test cases does not exceed 500 000.

Output

For each test case, if there is no valid rearrangement plan, output “No” in one line. Otherwise, output “Yes” in one line, followed by h lines in the same format as the input (without the line of h and w). These h lines should describe the information of the floor after the rearrangement. If there are multiple valid rearrangement plans, any of them is accepted.

Sample Input	Sample Output
3	Yes
3 4	^##.
.##.	v<>#
^<>#	<>#.
v.#.	No
3 3	Yes
##.	^.
.<>	v.
##.	
2 2	
..	
..	