## Codeforces Round #699 (Div. 2)

# A. Space Navigation

2 seconds, 256 megabytes

You were dreaming that you are traveling to a planet named Planetforces on your personal spaceship. Unfortunately, its piloting system was corrupted and now you need to fix it in order to reach Planetforces.

Space can be represented as the $XY$ plane. You are starting at point $(0, 0)$, and Planetforces is located in point $(p_x, p_y)$.

The piloting system of your spaceship follows its list of orders which can be represented as a string $s$. The system reads $s$ from left to right. Suppose you are at point $(x, y)$ and current order is $s_i$:

- if $s_i = \mathrm{U}$, you move to $(x, y + 1)$;
- if $s_i = \mathrm{D}$, you move to $(x, y - 1)$;
- if $s_i = \mathrm{R}$, you move to $(x + 1, y)$;
- if $s_i = \mathrm{L}$, you move to $(x - 1, y)$.

Since string $s$ could be corrupted, there is a possibility that you won't reach Planetforces in the end. Fortunately, **you can delete some orders from $s$ but you can't change their positions**.

Can you delete several orders (possibly, zero) from $s$ in such a way, that you'll reach Planetforces after the system processes all orders?

### Input

The first line contains a single integer $t$ ($1 \le t \le 1000$) — the number of test cases.

Each test case consists of two lines. The first line in each test case contains two integers $p_x$ and $p_y$ ($-10^5 \le p_x, p_y \le 10^5$; $(p_x, p_y) \neq (0, 0)$) — the coordinates of Planetforces $(p_x, p_y)$.

The second line contains the string $s$ ($1 \le |s| \le 10^5$: $|s|$ is the length of string $s$) — the list of orders.

It is guaranteed that the sum of $|s|$ over all test cases does not exceed $10^5$.

### Output

For each test case, print "YES" if you can delete several orders (possibly, zero) from $s$ in such a way, that you'll reach Planetforces. Otherwise, print "NO". You can print each letter in any case (upper or lower).

**input**

```
6
10 5
RRRRRRRRRRUUUUU
1 1
UDDDRLLL
-3 -5
LDLDLDDDR
1 2
LLLLUU
3 -2
RDULRLLDR
-1 6
RUDURUUUUR
```

**output**

```
YES
YES
YES
NO
YES
NO
```

In the first case, you don't need to modify $s$, since the given $s$ will bring you to Planetforces.

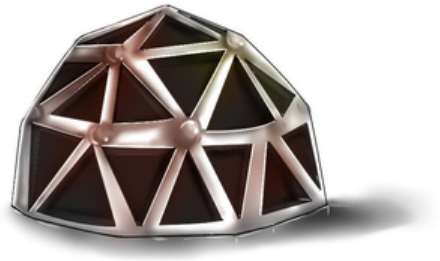
In the second case, you can delete orders $s_2$, $s_3$, $s_4$, $s_6$, $s_7$ and $s_8$, so $s$ becomes equal to "UR".

In the third test case, you have to delete order $s_9$, otherwise, you won't finish in the position of Planetforces.

# B. New Colony

2 seconds, 256 megabytes

After reaching your destination, you want to build a new colony on the new planet. Since this planet has many mountains and the colony must be built on a flat surface you decided to flatten the mountains using boulders (you are still dreaming so this makes sense to you).



You are given an array $h_1, h_2, \ldots, h_n$, where $h_i$ is the height of the $i$-th mountain, and $k$ — the number of boulders you have.

You will start throwing boulders from the top of the first mountain one by one and they will roll as follows (let's assume that the height of the current mountain is $h_i$):

- if $h_i \ge h_{i+1}$, the boulder will roll to the next mountain;
- if $h_i < h_{i+1}$, the boulder will stop rolling and increase the mountain height by $1$ ($h_i = h_i + 1$);
- if the boulder reaches the last mountain it will fall to the waste collection system and disappear.

You want to find the position of the $k$-th boulder or determine that it will fall into the waste collection system.

### Input

The first line contains a single integer $t$ ($1 \le t \le 100$) — the number of test cases.

Each test case consists of two lines. The first line in each test case contains two integers $n$ and $k$ ($1 \le n \le 100$; $1 \le k \le 10^9$) — the number of mountains and the number of boulders.

The second line contains $n$ integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 100$) — the height of the mountains.

It is guaranteed that the sum of $n$ over all test cases does not exceed $100$.

### Output

For each test case, print $-1$ if the $k$-th boulder will fall into the collection system. Otherwise, print the position of the $k$-th boulder.

**input**

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

**output**

```
2
1
-1
-1
```

Let's simulate the first case:

- The first boulder starts at $i = 1$; since $h_1 \geq h_2$ it rolls to $i = 2$ and stops there because $h_2 < h_3$.
- The new heights are $[4, 2, 2, 3]$.
- The second boulder starts at $i = 1$; since $h_1 \geq h_2$ the boulder rolls to $i = 2$; since $h_2 \geq h_3$ the boulder rolls to $i = 3$ and stops there because $h_3 < h_4$.
- The new heights are $[4, 2, 3, 3]$.
- The third boulder starts at $i = 1$; since $h_1 \geq h_2$ it rolls to $i = 2$ and stops there because $h_2 < h_3$.
- The new heights are $[4, 3, 3, 3]$.

The positions where each boulder stopped are the following: $[2, 3, 2]$.

In the second case, all $7$ boulders will stop right at the first mountain rising its height from $1$ to $8$.

The third case is similar to the first one but now you'll throw $5$ boulders. The first three will roll in the same way as in the first test case. After that, mountain heights will be equal to $[4, 3, 3, 3]$, that's why the other two boulders will fall into the collection system.

In the fourth case, the first and only boulders will fall straight into the collection system.

## C. Fence Painting

2 seconds, 256 megabytes

You finally woke up after this crazy dream and decided to walk around to clear your head. Outside you saw your house's fence — so plain and boring, that you'd like to repaint it.

You have a fence consisting of $n$ planks, where the $i$-th plank has the color $a_i$. You want to repaint the fence in such a way that the $i$-th plank has the color $b_i$.

You've invited $m$ painters for this purpose. The $j$-th painter will arrive at the moment $j$ and will recolor **exactly one** plank to color $c_j$. For each painter you can choose which plank to recolor, but you can't turn them down, i. e. each painter has to color exactly one plank.

Can you get the coloring $b$ you want? If it's possible, print for each painter which plank he must paint.

### Input

The first line contains one integer $t$ ($1 \leq t \leq 10^4$) — the number of test cases. Then $t$ test cases follow.

The first line of each test case contains two integers $n$ and $m$ ($1 \leq n, m \leq 10^5$) — the number of planks in the fence and the number of painters.

The second line of each test case contains $n$ integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$) — the initial colors of the fence.

The third line of each test case contains $n$ integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_i \leq n$) — the desired colors of the fence.

The fourth line of each test case contains $m$ integers $c_1, c_2, \ldots, c_m$ ($1 \leq c_j \leq n$) — the colors painters have.

It's guaranteed that the sum of $n$ doesn't exceed $10^5$ and the sum of $m$ doesn't exceed $10^5$ over all test cases.

### Output

For each test case, output "NO" if it is impossible to achieve the coloring $b$.

Otherwise, print "YES" and $m$ integers $x_1, x_2, \ldots, x_m$, where $x_j$ is the index of plank the $j$-th painter should paint.

You may print every letter in any case you want (so, for example, the strings "yEs", "yes", "Yes" and "YES" are all recognized as positive answer).

**input**

```
6
1 1
1
1
1
5 2
1 2 2 1 1
1 2 2 1 1
1 2
3 3
2 2 2
2 2 2
2 3 2
10 5
7 3 2 1 7 9 4 2 7 9
9 9 2 1 4 9 4 2 3 9
9 9 7 4 3
5 2
1 2 2 1 1
1 2 2 1 1
3 3
6 4
3 4 2 4 1 2
2 3 1 3 1 1
2 2 3 4
```

**output**

```
YES
1
YES
2 2
YES
1 1 1
YES
2 1 9 5 9
NO
NO
```

## D. AB Graph

2 seconds, 256 megabytes

Your friend Salem is Warawreh's brother and only loves math and geometry problems. He has solved plenty of such problems, but according to Warawreh, in order to graduate from university he has to solve more graph problems. Since Salem is not good with graphs he asked your help with the following problem.

You are given a complete directed graph with $n$ vertices without self-loops. In other words, you have $n$ vertices and each pair of vertices $u$ and $v$ ($u \neq v$) has both directed edges $(u, v)$ and $(v, u)$.

Every directed edge of the graph is labeled with a single character: either 'a' or 'b' (edges $(u, v)$ and $(v, u)$ may have different labels).

You are also given an integer $m > 0$. You should find a path of length $m$ such that the string obtained by writing out edges' labels when going along the path is a **palindrome**. The length of the path is the number of edges in it.

**You can visit the same vertex and the same directed edge any number of times**.

### Input

The first line contains a single integer $t$ ($1 \leq t \leq 500$) — the number of test cases.

The first line of each test case contains two integers $n$ and $m$ ($2 \leq n \leq 1000$; $1 \leq m \leq 10^5$) — the number of vertices in the graph and desirable length of the palindrome.

Each of the next $n$ lines contains $n$ characters. The $j$-th character of the $i$-th line describes the character on the edge that is going from node $i$ to node $j$.

Every character is either 'a' or 'b' if $i \neq j$, or '*' if $i = j$, since the graph doesn't contain self-loops.

It's guaranteed that the sum of $n$ over test cases doesn't exceed $1000$ and the sum of $m$ doesn't exceed $10^5$.

### Output

For each test case, if it is possible to find such path, print "YES" and the path itself as a sequence of $m + 1$ integers: indices of vertices in the path in the appropriate order. If there are several valid paths, print any of them.

Otherwise, (if there is no answer) print "NO".

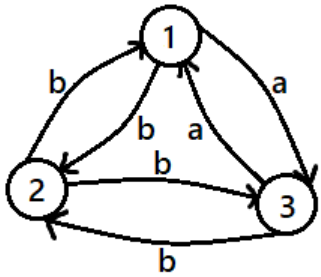
**input**

```
5
3 1
*ba
b*b
ab*
3 3
*ba
b*b
ab*
3 4
*ba
b*b
ab*
4 6
*aaa
b*ba
ab*a
bba*
2 6
*a
b*
```

**output**

```
YES
1 2
YES
2 1 3 2
YES
1 3 1 3 1
YES
1 2 1 3 4 1 4
NO
```

The graph from the first three test cases is shown below:




In the first test case, the answer sequence is $[1, 2]$ which means that the path is:

$$1 \xrightarrow{\text{b}} 2$$

So the string that is obtained by the given path is b.

In the second test case, the answer sequence is $[2, 1, 3, 2]$ which means that the path is:

$$2 \xrightarrow{\text{b}} 1 \xrightarrow{\text{a}} 3 \xrightarrow{\text{b}} 2$$

So the string that is obtained by the given path is bab.

In the third test case, the answer sequence is $[1, 3, 1, 3, 1]$ which means that the path is:

$$1 \xrightarrow{\text{a}} 3 \xrightarrow{\text{a}} 1 \xrightarrow{\text{a}} 3 \xrightarrow{\text{a}} 1$$

So the string that is obtained by the given path is aaaa.

The string obtained in the fourth test case is abaaba.

## E. Sorting Books

2 seconds, 256 megabytes

One day you wanted to read something, so you went to your bookshelf to grab some book. But when you saw how messy the bookshelf was you decided to clean it up first.

There are $n$ books standing in a row on the shelf, the $i$-th book has color $a_i$.

You'd like to rearrange the books to make the shelf look beautiful. The shelf is considered *beautiful* if all books of the same color are next to each other.

In one operation you can take one book from any position on the shelf and move it to the right end of the shelf.

What is the minimum number of operations you need to make the shelf beautiful?

### Input

The first line contains one integer $n$ ($1 \leq n \leq 5 \cdot 10^5$) — the number of books.

The second line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$) — the book colors.

### Output

Output the minimum number of operations to make the shelf beautiful.

| input |
| --- |
| 5<br>1 2 2 1 3 |
| **output** |
| 2 |

| input |
| --- |
| 5<br>1 2 2 1 1 |
| **output** |
| 1 |

In the first example, we have the bookshelf $[1, 2, 2, 1, 3]$ and can, for example:

1. take a book on position $4$ and move to the right end: we'll get $[1, 2, 2, 3, 1]$;
2. take a book on position $1$ and move to the right end: we'll get $[2, 2, 3, 1, 1]$.

In the second example, we can move the first book to the end of the bookshelf and get $[2, 2, 1, 1, 1]$.

## F. AB Tree

2 seconds, 256 megabytes

Kilani and Abd are neighbors for 3000 years, but then the day came and Kilani decided to move to another house. As a farewell gift, Kilani is going to challenge Abd with a problem written by their other neighbor with the same name Abd.

The problem is:

You are given a connected tree rooted at node $1$.

You should assign a character a or b to every node in the tree so that the total number of a's is equal to $x$ and the total number of b's is equal to $n - x$.

Let's define a string for each node $v$ of the tree as follows:

- if $v$ is root then the string is just one character assigned to $v$:
- otherwise, let's take a string defined for the $v$'s parent $p_v$ and add to the end of it a character assigned to $v$.

You should assign every node a character in a way that **minimizes the number of distinct strings** among the strings of all nodes.

### Input

The first line contains two integers $n$ and $x$ ($1 \le n \le 10^5$; $0 \le x \le n$) — the number of vertices in the tree the number of a's.

The second line contains $n - 1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i \le n$; $p_i \neq i$), where $p_i$ is the parent of node $i$.

It is guaranteed that the input describes a connected tree.

### Output

In the first line, print the minimum possible total number of distinct strings.

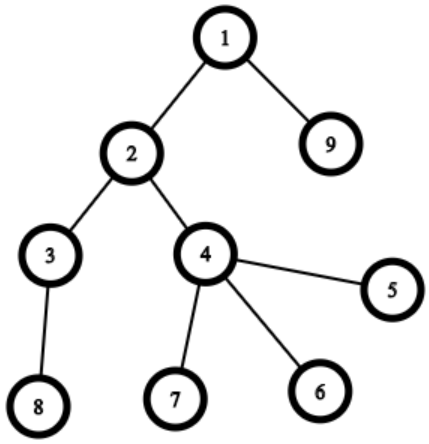
In the second line, print $n$ characters, where all characters are either a or b and the $i$-th character is the character assigned to the $i$-th node.

Make sure that the total number of a's is equal to $x$ and the total number of b's is equal to $n - x$.

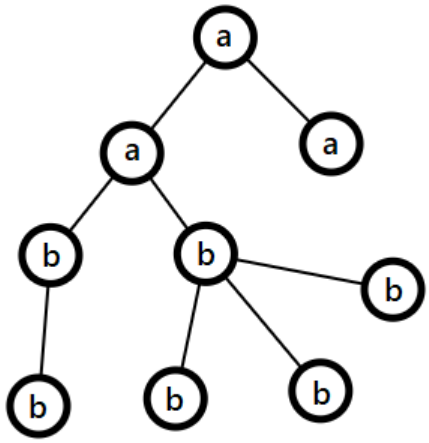
If there is more than one answer you can print any of them.

| input |
| --- |
| 9 3<br>1 2 2 4 4 4 3 1 |
| **output** |
| 4<br>aabbbbbba |

The tree from the sample is shown below:




The tree after assigning characters to every node (according to the output) is the following:




Strings for all nodes are the following:

- string of node $1$ is: a
- string of node $2$ is: aa
- string of node $3$ is: aab
- string of node $4$ is: aab
- string of node $5$ is: aabb
- string of node $6$ is: aabb
- string of node $7$ is: aabb
- string of node $8$ is: aabb
- string of node $9$ is: aa

The set of unique strings is $\{\text{a}, \text{aa}, \text{aab}, \text{aabb}\}$, so the number of distinct strings is $4$.