Problem A: Tower Defense

- Time Limit: 2 sec

Problem Statement

There are $M+1$ cells labeled from 0 to M arranged in sequence. Cell 0 contains a base, and cell M contains a monster with health H. There are N soldiers deployed near the cells. The attack range of soldier i is from cell L_i to R_i (i.e., cells $L_i, L_i+1, \ldots, R_i$).

Each turn, both the monster and the soldiers perform the following actions in order (first the monster, then the soldiers):

- Monster: If the monster's health is 1 or more and it is not in cell 0, it moves one cell closer to the base (i.e., it moves to the cell with a number 1 smaller).
- Soldiers: Any soldier whose attack range includes the cell where the monster is currently located attacks the monster and reduces its health by 1.

If the monster's health drops to 0 or below before it reaches cell 0, the monster dies in that cell, the defense is successful, and the game ends. If the monster reaches cell 0 without its health dropping to 0 or below, the defense fails and the game ends.

Determine whether the defense will succeed, and if so, find the number of the cell where the monster will die.

Input

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

$$
\begin{aligned}
&N\ M\ H\\
&L_1\ R_1\\
&\vdots\\
&L_N\ R_N
\end{aligned}
$$

The first line contains three integers, N, M, and H, representing the number of soldiers, the number of cells, and the monster's health, respectively. N is between 1 and $100{,}000$ (both inclusive). M is between 2 and 10^6 (both inclusive). H is between 1 and 10^{11} (both inclusive).

The next N lines each contain two integers, L_i and R_i, which represent the attack range interval of the i-th soldier. Both L_i and R_i are between 1 and $M-1$ (both inclusive). It is guaranteed that L_i is less than or equal to R_i.

Output

Output a single integer, the number of the cell where the monster will die if the defense is successful; otherwise, output -1.

Sample Input 1

```
2 5 3
2 4
3 4
```

Sample Output 1

```
3
```

Sample Input 2

```
4 5 10
1 2
2 4
4 4
3 4
```

Sample Output 2

```
-1
```

Problem B: Renovation

- Time Limit: 2 sec

Problem Statement

Jack is planning to renovate his house to make his room as large as possible. However, since Jack doesn't have much money, he wants to create a spacious room with minimal effort.

Jack's house is represented as a grid with height H and width W. Each cell in the grid is in one of the following states:

- `.` : A floor that Jack can freely move through.
- `#` : A wall that Jack cannot pass through.
- `S` : The cell where Jack is currently located, which is also a floor.

Jack can move to adjacent floor cells in the grid, either up, down, left, or right. He cannot move outside the boundaries of his house.

In this renovation, Jack can destroy up to one wall and turn it into a floor. Determine the maximum number of cells Jack can reach from the currently located position after making this change.

Input

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

$$
\begin{aligned}
&H\ W\\
&l_1\\
&l_2\\
&\vdots\\
&l_H
\end{aligned}
$$

The first line contains two integers, H and W, representing the height and width of the grid, respectively. Both H and W are between 2 and 500 (both inclusive).

The next H lines each contain a string of length W, representing the grid. Each string l_i consists of the characters `.` (floor), `#` (wall), and `S` (Jack's starting position). It is guaranteed that the grid contains exactly one `S`.

Output

Output a single integer, the maximum number of cells Jack can reach from his starting position after optionally destroying one wall.

Sample Input 1

```
3 5
.#...
.####
#.#.S
```

Sample Output 1

```
6
```

Sample Input 2

```
3 7
.......
...S...
.......
```

Sample Output 2

```
21
```

Problem C: Commutativity

- Time Limit: 2 sec

Problem Statement

You are given a function $F : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$. In other words, F is a function that takes an integer x between 1 and N inclusive and returns an integer $F(x)$ between 1 and N inclusive. Your task is to count the number of functions $G : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ such that F and G commute under composition: that is, $F(G(x)) = G(F(x))$ holds for any $x \in \{1, 2, \ldots, N\}$.

As this number could be large, print the answer modulo $998{,}244{,}353$.

Input

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

N
$F_1\ F_2\ \ldots\ F_N$

The first line consists of an integer N between 1 and $5{,}000$, inclusive. The second line consists of N positive integers $F_1, F_2, \ldots, F_N$. For each i $(1 \leq i \leq N)$, F_i represents the value of $F(i)$. It is guaranteed that $1 \leq F_i \leq N$.

Output

Print the number of possible functions $G : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ modulo $998{,}244{,}353$.

Sample Input 1	Sample Output 1
5 4 5 3 2 1	5

Sample Input 2	Sample Output 2
8 2 3 1 3 6 7 5 5	64

Sample Input 3	Sample Output 3
10 3 1 4 1 5 9 2 6 5 3	64

Sample Input 4	Sample Output 4
15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	567381138

Problem D: Paper Cut Game

- Time Limit: 2 sec

Problem Statement

There is a sheet of paper consisting of a grid with N rows and M columns. Two players participate in a game with this paper.

Each player alternates moves, performing exactly one of the following actions on their turn.

- Cut the paper vertically along one of the grid lines to split the paper into two, and keep only the part with more cells. If both parts have the same number of cells, keep only one of them.
- Cut the paper horizontally along one of the grid lines to split the paper into two, and keep only the part with more cells. If both parts have the same number of cells, keep only one of them.

The player who is unable to make a move loses, while the other player wins. Given the size of the paper, determine the winner, assuming both players play optimally.

You have T test cases to solve.

Input

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

$$
\begin{aligned}
&T\\
&\mathrm{case}_1\\
&\vdots\\
&\mathrm{case}_T
\end{aligned}
$$

The first line contains an integer T between 1 and $200{,}000$, inclusive.

The $(i+1)$-th line corresponds to the i-th test case. Each line contains two integers, N and M, representing the height and width of the paper, respectively. Both N and M are between 1 and 10^{18}, inclusive.

Output

Print T lines. For each test case, the i-th line should contain `First` if the first player wins in the i-th test case, and `Second` otherwise.

Sample Input	Sample Output
5 1 1 1 2 3 3 7 3 999999999999999999 1000000000000000000	Second First Second Second First

Problem E: Ball Passing

- Time Limit: 2 sec

Problem Statement

There are M balls of each color $1, 2, \ldots, N$, and N students are sitting in a circle playing with these balls. The students are numbered from 1 to N in the order in which they are seated.

Initially, each student has exactly M balls. Specifically, the color of the j-th ball that i-th student has is color $a_{i,j}$. They will perform the following operation to achieve a state where each student holds balls of only one color:

Operation: Each of the N students simultaneously passes one ball they currently possess to their neighbor (the i-th student passes a ball to the $(i+1)$-th student, and the N-th student passes a ball to the 1st student, as the $(N+1)$-th student is considered to be the 1st student).

Your task is to determine whether, by repeating this operation at most NM times, it is possible to achieve the goal. If possible, output a sequence of operations.

Input

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

$$
\begin{aligned}
&N\ M \\
&a_{1,1}\ a_{1,2}\ \ldots\ a_{1,M} \\
&a_{2,1}\ a_{2,2}\ \ldots\ a_{2,M} \\
&\vdots \\
&a_{N,1}\ a_{N,2}\ \ldots\ a_{N,M}
\end{aligned}
$$

The first line contains two integers N and M, where N represents the number of students and M represents the number of balls of each color. Both N and M are between 2 and 100 (for N) and 2 and 100 (for M), inclusive.

Each of the following N lines consists of M positive integers $a_{i,1}, a_{i,2}, \ldots, a_{i,M}$. The integer $a_{i,j}$ represents the color of the j-th ball that the i-th student initially has. It is guaranteed that $1 \leq a_{i,j} \leq N$ and that each integer $1, 2, \ldots, N$ occurs M times among $a_{i,j}$ $(1 \leq i \leq N, 1 \leq j \leq M)$.

It is also guaranteed that the initial state does not satisfy the goal condition.

Output

Print -1 if it is impossible to achieve the goal state by at most NM operations.

Otherwise, print K — the number of operations. In the following K lines, print N integers $c_{i,1}, c_{i,2}, \ldots, c_{i,N}$. Here, $c_{i,j}$ represents the color of the ball passed from the j-th student to the $(j+1)$-th student at the i-th operation.

Sample Input 1

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

Sample Output 1

```
2
1 2
1 2
```

Sample Input 2

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

Sample Output 2

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

Problem F: Halfway Through the Book

- Time Limit: 2 sec

Problem Statement

A book titled “Immense Catalog of Permutation Compilation” is published by JAG Publisher. This book is extremely lengthy, with a total of $2^N - 1$ pages. Each page contains one of the non-empty subsequences (not necessarily contiguous) of a permutation P of length N (i.e., a rearrangement of $(1, \ldots, N)$). Each subsequence appears exactly once in lexicographical order. In other words, on the page k, you’ll find the k-th lexicographically smallest subsequence of P among all non-empty subsequences.

You’ve tried to read the entire book but gave up. However, you want to impress your friends by claiming that you read half of it, so you need to find the sequence on the exact middle page, which is page 2^{N-1}. Your task is to determine this sequence.

Input

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

N
$P_1\ P_2\ \ldots\ P_N$

The first line contains an integer N, where N represents the length of permutation P. N is between 1 and $10{,}000$. The second line contains N integers $P_1, \ldots, P_N$ ($1 \leq P_i \leq N$) which represent the permutation P.

Output

On the first line, print M, which represents the length of the sequence on page 2^{N-1}. On the second line, print M integers $Q_1, Q_2, \ldots, Q_M$, which represent the subsequence on page 2^{N-1}.

Sample Input 1	Sample Output 1
3 2 1 3	2 2 1

Sample Input 2	Sample Output 2
6 3 6 2 1 5 4	4 3 6 1 5

Problem G: Interactive String Guessing

- Time Limit: 2 sec

Problem Statement

This is an interactive problem.

Your task is to write a program that guesses a secret string through repetitive queries and responses. The secret string consists of **a** and **b**, and its length is between 1 and 1,000. Let S be the secret string. In a query, you specify a string T. In response to this, whether T is a substring of S or not will be returned. Here, T is said to be a substring of S when some contiguous part of S matches T. For example, **aba** is a substring of **babab** and **aba** but not of **abba**. You can make up to 1,100 queries.

Interaction

You should start with sending a query to the standard output and receiving the response from the standard input. This interaction can be repeated a number of times. When you have become confident of your guess through these interactions, you can send your answer. A query should be in the following format, followed by a newline.

> **?** T

Here, T is a non-empty string consisting of **a** and **b**, and its length is between 1 and 1,000, inclusive. The response to the query is given from the standard input in the following format followed by a newline.

> R

Here, R denotes the response to the query. R is **Yes** when T is a substring of the secret string S, and R is **No** when T is not a substring of the secret string S. However, if T does not satisfy the constraints, or the number of queries exceeds the limit, then R is **Invalid**. If the judge returns **Invalid**, your program is already considered incorrect, so terminate the program immediately.

You should send your answer to the standard output in the following format, followed by a newline.

> **!** T

Here, T is the secret string you have identified, its length is between 1 and 1,000, inclusive. You can send the answer only once, so you have to become sure of your guess through repeated interactions before sending the answer. After sending the answer, your program should terminate without any extra output.

Notes on interactive judging

The verdict will be indeterminate if there is malformed output during the interaction or your program quits prematurely. Terminate the program immediately after printing the answer, or the verdict will be indeterminate. As some environments require flushing the output buffers, make sure that your outputs are actually sent. Otherwise, your outputs will never reach the judge.

The string S will be fixed before the first interaction and will not be changed according to your questions or other factors.

Sample Interaction

Read	Write
	`? aba`
`Yes`	
	`? baa`
`No`	
	`! abab`

Problem H: Coins on a Tree

- Time Limit: 5 sec

Problem Statement

There is an undirected tree with N vertices numbered through 1 to N. There is a coin on each vertex, and a lowercase English letter is written on each coin.

You are playing a game to collect all the N coins. You have a string S which is initially empty. You first visit the vertex 1 and then repeatedly move to an adjacent vertex. You may visit each vertex any number of times. Whenever you visit a vertex where a coin is still put, you collect the coin and append the letter on the coin to the end of S. When there is already no coin on the node you visit, you do nothing. Note that the coin on the vertex 1 is collected first.

After collecting all the N coins, you have a string S of length N. What is the lexicographically smallest possible string that S can become?

Input

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

N
$p_2\ p_3\ \cdots\ p_N$
$c_1c_2\ldots c_N$

The first line consists of an integer N ($2 \leq N \leq 200{,}000$), which is the number of vertices on the tree.

The second line consists of $N-1$ integers. Each p_i ($2 \leq i \leq N$, $1 \leq p_i < i$) represents that the vertices i and p_i are adjacent. Note that p_1 is not given.

The third line consists of a string of N lowercase English letters. The i-th letter c_i ($1 \leq i \leq N$) is the letter on the coin on the vertex i.

Output

Print the lexicographically smallest possible string that S can become after collecting all the N coins.

Sample Input 1	Sample Output 1
7 1 1 2 2 3 3 abbabac	abababc

Sample Input 2	Sample Output 2
8 1 1 3 4 3 1 7 icpcicpc	icpccipc

Problem I: Fragile Tree

- Time Limit: 2 sec

Problem Statement

In the ICPC Park, there is a giant tree where a group of squirrels has made their nests.

The squirrels' nests consist of N rooms numbered $1, 2, \ldots, N$ and $N-1$ bidirectional roads numbered $1, 2, \ldots, N-1$ that connect the rooms. It is guaranteed that any two rooms can be reached from each other by following the roads.

Room 1 is the gathering place for the group, while the other rooms serve as sleeping quarters. Each room i has a_i squirrels living in it, where $a_1 = 0$.

Now, the squirrels are trying to gather in Room 1 for a morning assembly by following the roads from their sleeping quarters. However, due to a severe storm last night, each road has become fragile, and the i th road can only allow up to c_i squirrels to pass through before it collapses.

As an animal lover, your task is to choose one road to reinforce in order to maximize the number of squirrels that can reach Room 1. The reinforced road will not collapse, no matter how many squirrels pass through it.

Input

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

$$
\begin{aligned}
&N \\
&a_1\ a_2\ \ldots\ a_N \\
&u_1\ v_1\ c_1 \\
&u_2\ v_2\ c_2 \\
&\vdots \\
&u_{N-1}\ v_{N-1}\ c_{N-1}
\end{aligned}
$$

The first line consists of an integer N between 2 and $200{,}000$, inclusive.

The second line contains N non-negative integers $a_1, a_2, \ldots, a_N$. The integer a_i denotes the number of squirrels living in Room i, with $a_1 = 0$ and $0 \leq a_i \leq 10^9$ $(i = 2, 3, \ldots, N)$.

Each of the following $N-1$ lines contains three integers u_i, v_i and c_i ($1 \leq u_i, v_i \leq N$, $0 \leq c_i \leq 10^9$). u_i and v_i denote the endpoints of the i th road, and c_i denotes its durability.

It is guaranteed that the given graph is a tree.

Output

Print the maximum number of squirrels that can reach Room 1.

Sample Input	Sample Output
8 0 11 13 14 17 19 15 12 1 2 5 2 3 2 3 4 3 1 5 2 5 6 5 1 7 4 7 8 3	31

Problem J: Draw the Tree

- Time Limit: 3 sec

Problem Statement

You are given a tree of N vertices, numbered $1, 2, \ldots, N$. You want to determine a positive integer value M and draw this tree on a region $R = \{(x, y) \mid 0 \leq x \leq M - 1, 0 \leq y \leq 1\}$ on a two-dimensional coordinate plane.

In this drawing, the vertices of the tree should be placed at grid points within R, and the edges of the tree should be drawn as straight line segments. Furthermore, these line segments representing different edges of the tree should not share any points other than their endpoints.

More formally, you want to construct a mapping p from the vertex set of tree T to the set of grid points $\{(x, y) \mid x \in \{0, 1, \ldots, M - 1\}, y \in \{0, 1\}\}$ such that the following properties are satisfied:

- For any distinct vertices v_i and v_j in T, $p(v_i)$ and $p(v_j)$ are distinct.
- For any distinct edges $e_i = (u_i, v_i)$ and $e_j = (u_j, v_j)$ in T, the line segments $\mathbf{seg}_i$ connecting $p(u_i)$ and $p(v_i)$, and $\mathbf{seg}_j$ connecting $p(u_j)$ and $p(v_j)$ do not share any points inside $\mathbf{seg}_i$ (excluding the endpoints) or inside $\mathbf{seg}_j$ (excluding the endpoints).

Your task is to determine if such a drawing is possible by choosing an appropriate M, and if possible, find the minimum value of M.

Input

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

$$
\begin{aligned}
&N \\
&u_1\ v_1 \\
&u_2\ v_2 \\
&\vdots \\
&u_{N-1}\ v_{N-1}
\end{aligned}
$$

The first line consists of an integer N, which is between 1 and $50{,}000$, inclusive.

Each of the following $N - 1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq N$). u_i and v_i denote the endpoints of the i-th edge of T.

It is guaranteed that the given graph is a tree.

Output

Output a single integer — the minimum possible value of M to achieve the goal. If it is impossible to do so, output -1 as the answer.

Sample Input 1	Sample Output 1
6 1 2 1 3 1 4 1 5 1 6	3

Sample Input 2

```
21
1 2
1 3
1 4
1 5
6 7
6 8
6 9
6 10
11 12
11 13
11 14
11 15
16 17
16 18
16 19
16 20
5 21
10 21
15 21
20 21
```

Sample Output 2

```
-1
```

Sample Input 3

```
18
1 2
2 3
2 4
2 5
1 6
6 7
6 8
6 9
1 12
10 11
11 12
12 13
13 14
1 16
15 16
16 17
17 18
```

Sample Output 3

```
11
```

Sample Input 4

```
21
20 10
7 2
7 17
18 5
3 1
15 17
11 7
14 18
11 21
6 2
8 19
17 14
4 18
16 17
9 10
19 17
18 12
15 3
13 15
16 10
```

Sample Output 4

```
11
```

Problem K: Equal or Not Equal

- Time Limit: 3 sec

Problem Statement

There are N integer variables $a_1, \ldots, a_N$. All the initial values are unspecified. You have to process M events in order, each of which is an observation event, a change event or an inquiry event.

One kind of an observation event has the format,

```
1 x y
```

which represents that it is observed that a_x and a_y are equal. In other words, after this event it is guaranteed that these two variables have the same value unless there is a change event for either variable.

The other kind of an observation event has the format,

```
2 x y
```

which represents that it is observed that a_x and a_y are not equal.

A change event has the format,

```
3 x
```

which represents that the value of a_x changes to a different integer.

Finally, an inquiry event has the format,

```
4 x y
```

which asks whether a_x and a_y are equal. If it can be proven from all the past events that a_x and a_y are equal, you have to print **Yes**. Similarly, if it can be proven that a_x and a_y are not equal, you have to print **No**. If neither can be proven, you have to print **Unknown**.

Input

The input consists of a single test case of the following format, where all values in the input are integers.

$$
\begin{aligned}
&N\ M \\
&\mathrm{event}_1 \\
&\vdots \\
&\mathrm{event}_M
\end{aligned}
$$

The integer N is the number of variables ($1 \leq N \leq 10^6$). The integer M is the number of events ($1 \leq M \leq 500{,}000$).

M events are given in chronological order. The format of each event is explained above. In an observation event or an inquiry event, it is guaranteed that $1 \leq x < y \leq N$. In a change event it is guaranteed that $1 \leq x \leq N$.

At any point, it is guaranteed that there is at least one assignment to all N variables that contradicts none of the given information.

Output

For each inquiry event, print **Yes**, **No** or **Unknown** followed by a newline.

Sample Input	Sample Output
4 11 1 1 2 4 1 2 2 3 4 4 3 4 4 1 3 1 1 3 4 1 3 4 1 4 3 1 4 1 3 4 1 4	Yes No Unknown Yes No No Unknown