





Czech ACM Student Chapter
CTU FEE Prague, MFF UK Prague, FEI TU Ostrava
FI MU Brno, FIIT STU Bratislava

CTU Open 2006





Tree Marking Animals

no files available yet

A *tree marker* is an exotic animal that likes numbers and lives on rooted trees. A rooted tree has several nodes, one of them is marked as root. From each node lead several branches to other nodes; one of the branches is a downward branch. There is no downward branch from the root, and each branch of the tree is downward from some node. By following the downward branches from any node, we eventually always reach the root of the tree.

Let us return to tree markers. Tree markers climb through the tree randomly, performing the following operations: *Mark* the node - they leave a mark in the node in that they currently are. *Unmark* the node - they remove all the marks from their current node (or do nothing if no mark is there). And finally, *jump* - the tree marker jumps and falls downward through the tree, until it hits a marked node (possibly the same node it jumped from) or the root of the tree. Your task is to write program that simulates these fascinating creatures.

Input Specification

The input consists of several instances separated by single empty lines; input of each instance consists of several lines. Each input instance is finished by a line containing a string "All tree markers went for lunch, they will be back at 15:00.". The first line of each instance contains one integer n ($1 \leq n \leq 1000000$), number of nodes of the tree. The nodes of the tree are numbered from 1 to n, where 1 is the root of the tree. The second line of the instance consists of n - 1 numbers a_2, ..., a_n ($1 \leq a_i < i$) - a_i is the number of the node to that the downward branch from the i-th node leads. The remaining lines (except for the last one) contain description of one of the events:

- A string "Mark" followed by an integer k ($1 \leq k \leq n$) - a tree marker marks the node k.
- A string "Unmark" followed by an integer k ($1 \leq k \leq n$) - a tree marker unmarks the node k.
- A string "Jump" followed by an integer k ($1 \leq k \leq n$) - a tree marker jumps from the node k.

Output Specification

For each input instance, output a single line. This line contains several integers separated by single spaces - for each jump event in the instance, write the number of the node in that the tree marked ends his jump.

Sample Input

```
5
1 2 3 4
Mark 2
Mark 3
Jump 5
Mark 4
Jump 5
Unmark 2
Jump 5
Unmark 4
Unmark 3
Jump 5
Mark 4
Jump 4
Unmark 4
Jump 4
All tree markers went for lunch, they will be back at 15:00.

1

Jump 1
Mark 1
Jump 1
All tree markers went for lunch, they will be back at 15:00.
```

Output for Sample Input

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





Black Boxes

no files available yet

A *pattern of length k* is a sequence $a_1, a_2, ..., a_k$ of k pairwise distinct numbers. Two patterns $a_1, a_2, ..., a_k$ and $b_1, b_2, ..., b_k$ are equivalent if $a_i < a_j \Leftrightarrow b_i < b_j$ for all i and j such that $1 \le i < j \le k$. A *dictionary* is a list of several pairwise nonequivalent patterns.

A *black box* for a fixed dictionary is a machine that recognizes patterns - given an unknown pattern X, it asks several questions of form "Is the i-th element of the pattern smaller than the j-th one?", and then decides to which of the patterns in the dictionary is X equivalent (in case X is not equivalent to any of the patterns, the black box may answer arbitrarily). The *quality* of the black box is the maximum number of questions the black box asks for any input.

International Blackbox Machine Corporation manufactures black boxes for various dictionaries, and claims that they ask the least number of questions possible. Your task is to help to verify this claim.

Input Specification

The input consists of several instances, the instances are separated by single empty lines. The first line of each instance contains integers D and k ($1 \leq D \leq 120, 1 \leq k \leq 100$) separated by a space, where D is the number of patterns in the dictionary and k is the length of each pattern. Each of the D following lines describes one of the patterns in the dictionary. Each of the lines consist of k distinct integers $a_1, ..., a_k$ ($1 \leq a_i \leq k$) separated by single spaces.

Output Specification

For each input instance, output a single line containing a single integer Q - the quality of the best possible black box that recognizes the given dictionary.

Sample Input

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

1 1
1
```

Output for Sample Input

```
2
0
```

Comment: It is easy to see that no black box may recognize these patterns by just one question. The black box that recognizes them by two questions may work for example this way: It first asks whether the first element a_1 is smaller than the second one a_2. If this is the case, it asks whether the second element is smaller than the third one. On the other hand, if $a_1 > a_2$, it asks whether the third element is smaller than the fourth one.





Chocolate Eating

no files available yet

Alice and Bob play the following game: they start with several chocolate bars, consisting of n_1 , n_2 , ... , n_k pieces. The players alternate on turns, and Alice starts. In each turn, the player may either split one of the bars into two (i.e., replace a bar consisting of $n > 1$ pieces with two bars consisting of a and b pieces, for any positive integers a and b such that $n = a + b$), or eat a bar of chocolate that consist of only one piece (if such a piece is available). The game ends when there is no chocolate left. Obviously, the goal of the game is to eat as much chocolate as possible.