

Long: WCWBTT

Input file: *standard input*
Output file: *standard output*
Time limit: 15 seconds
Memory limit: 1024 mebibytes

This is an interactive problem.

You need to maintain a rooted tree with vertex 1 as the root. The tree has n vertices, and the parent of vertex i ($2 \leq i \leq n$) is p_i ($1 \leq p_i < i$).

You have to process q queries, each in one of the following forms:

- “? a b ”: Output the set S , where S is the set consisting of all vertices on the unique simple path from a to b .
- “= a b ”: Change the parent of a to b (that is, $p_a \leftarrow b$). It is guaranteed that the vertices still form a tree after the modification, but it is **not guaranteed that** $b < a$.

But you soon discover a problem: the size of the set S may be too large, you can't output all elements in each query.

To deal with this issue, you have designed a special computer. This special computer can maintain sets of integers and operate on them quickly. Initially, the computer has only $n + 1$ sets S_0, S_1, \dots, S_n where the set $S_0 = \emptyset$, and $S_i = \{i\}$ for all $1 \leq i \leq n$.

This computer is efficient and at the same time very simple: it supports only two different operations!

- “+ a b ”: Construct a new set $S_c = S_a \cup S_b$ ($S_a \cap S_b = \emptyset$), with c being the maximum of the ID of all sets plus one. **You have to make sure that** $S_a \cap S_b = \emptyset$. The cost of this operation is $|S_a| + |S_b|$.
- “! k x_1 x_2 ... x_k ”: Print the set $S_{x_1} \cup S_{x_2} \cup \dots \cup S_{x_k}$ as the answer to the query. **You need to ensure that** $S_{x_i} \cap S_{x_j} = \emptyset$ for all $1 \leq i < j \leq k$. The cost of this operation is k .

Now, you need to use this computer to maintain the rooted tree. In order to avoid calculations consuming too much time and causing damage to the computer, there are the following restrictions when using the computer.

- The cost of **each operation** cannot exceed 7000.
- The **sum** of the cost of all operations cannot exceed $7.5 \cdot 10^7$.
- The total number of operations cannot exceed $5 \cdot 10^6$.

Input

The first line of the input contains two integers n and q ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq q \leq 2.5 \cdot 10^4$).

The next line of the input contains $n - 1$ integers p_2, p_3, \dots, p_n ($1 \leq p_i < i$), indicating the initial parent of each vertex.

Interaction Protocol

The interaction library will then perform q queries. Each query will contain one line in either the form “? a b ” or the form “= a b ”. These forms are described above. After each query, you can perform as many operations as you want.

In particular, you need to perform operation “! k ...” exactly once after the query “= a b ”. After each operation “! k ...”, you need to flush your output.

Please note that, due to the huge output, we recommend you to flush the output **only after** each operation “! k ...”.

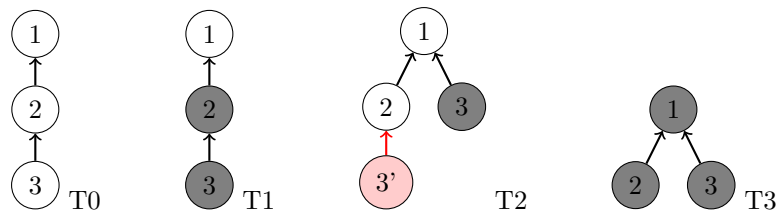
Example

<i>standard input</i>	<i>standard output</i>
3 3	
1 2	
? 2 3	+ 1 2
	+ 3 4
	! 2 2 3
	<flush the output>
= 3 1	
	+ 1 2
	+ 1 3
? 2 3	
	! 2 2 7
	<flush the output>

Note

The sample input and output are intended only to illustrate the interaction protocol. The string “<flush the output>” and the blank lines are only added for the reader’s convenience. You should not output this information.

Here’s the figure of the sample test case:



The figure corresponds to the sample test case

- $S_0 = \emptyset$
- $S_1 = \{1\}$
- $S_2 = \{2\}$
- $S_3 = \{3\}$
- $S_4 = \{1, 2\}$
- $S_5 = \{1, 2, 3\}$
- $S_6 = \{1, 2\}$
- $S_7 = \{1, 3\}$