Kikai No Tsubasa

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

This is a communication problem.

Teto is studying the following NP-complete problem:

- You are given an undirected simple graph G with n vertices and m edges. The vertices are numbered from 1 to n.
- A permutation $p = [p_1, p_2, \cdots, p_n]$ of $1, 2, \cdots, n$ is a **valid labeling** of the graph if and only if $p_u + p_v \leq n + 1$ for every edge (u, v).
- It is guaranteed that at least one valid labeling exists. You need to find any valid labeling of the given graph.

In the year 6202, *Teto* not only invented an efficient algorithm for this problem with the help of a quantum computer, but also invented a time machine! Now, *Teto* wants to bring a solution to this problem back in time and accelerate the research into the algorithm. However, she finds that she cannot remember too much information: besides remembering the graph exactly, she can remember only $\mathbf{5 \times 10^5}$ additional bits. Can you help her compress the information in the solution?

More specifically, you need to design a program that, given G and p, can generate a **non-empty** binary string s of length at most 5×10^5, and can then independently find a valid labeling p' using only G and s. Note that p' only needs to be a valid labeling of the graph and does not have to be identical to p.

To test your program, the judging system will run the same program twice on each test, corresponding to the two processes described above. The two runs do not share memory, files, or any other state. Apart from the same graph and the binary string output on the first run, the second run receives no information from the first run.

Input

The first line of each run contains a string op ($op \in \{\texttt{first}, \texttt{second}\}$), denoting which run of the program it is.

The second line contains two integers n and m ($2 \leq n \leq 10^5$, $0 \leq m \leq 5 \times 10^5$), denoting the number of vertices and the number of edges, respectively. It is guaranteed that n is even.

The next m lines follow. The i-th of these lines contains two integers u_i, v_i ($1 \leq u_i < v_i \leq n$), denoting an edge connecting vertices u_i and v_i.

If $op = \texttt{first}$, the last line contains n integers $p_1, p_2, \cdots, p_n$, representing a valid labeling given to the program.

If $op = \texttt{second}$, the last line contains a **non-empty** binary string s, representing the information output on the first run. It is guaranteed that **the graph and the order of its edges are identical in both runs.**

It is guaranteed that all edges in the graph are distinct. If $op = \texttt{first}$, it is also guaranteed that p is a permutation of $1, 2, \cdots, n$ and that $p_u + p_v \leq n + 1$ for every edge (u, v).

Interaction Protocol

If $op = \texttt{first}$, output a **non-empty** string s consisting only of the characters `0` and `1`, where $|s| \leq 5 \times 10^5$. Do not output anything else.

If $op = \texttt{second}$, output a permutation $p'_1, p'_2, \cdots, p'_n$ of $1, 2, \cdots, n$ such that $p'_u + p'_v \leq n + 1$ for every edge (u, v).

Examples

standard input	standard output
first 8 7 1 5 2 6 3 7 4 8 1 2 2 3 3 4 1 2 3 4 8 7 6 5	10011010100101000100110 10
second 8 7 1 5 2 6 3 7 4 8 1 2 2 3 3 4 10011010100101000100110 10	4 1 3 2 5 8 6 7