Problem D. DAG Serialization

Time limit: 3 seconds
Memory limit: 1024 megabytes

Consider a simple single-bit boolean register that supports two operations:

- **set** — sets the register to **true** if it was **false**, and returns **true**; otherwise, it returns **false**;
- **unset** — sets the register to **false** if it was **true**, and returns **true**; otherwise, it returns **false**.

The initial state of the register is **false**. Suppose there were n operations op_i (for $1 \le i \le n$) where **at most two operations returned true**. Also, we are given the partial order of operations as a directed acyclic graph (DAG): an edge $i \to j$ means that op_i happened before op_j. You are asked whether it is possible to put these operations in some linear sequential order that satisfies the given partial order and such that if operations are applied to the register in that order, their results are the same as given.

Input

In the first line, you are given an integer n — the number of operations ($1 \le n \le 10^5$). In the following n lines, you are given operations in the format "*type result*", where *type* is either "`set`" or "`unset`" and *result* is either "`true`" or "`false`". It is guaranteed that at most two operations have "`true`" results.

In the next line, you are given an integer m — the number of arcs of the DAG ($0 \le m \le 10^5$). In the following m lines, you are given arcs — pairs of integers a and b ($1 \le a, b \le n$; $a \ne b$). Each arc indicates that operation op_a happened before operation op_b.

Output

Print any linear order of operations that satisfies the DAG constraints and ensures the results of the operations match the ones given in the input. If a correct operation order does not exist, print -1.

Examples

standard input	standard output
5 set true unset true set false unset false unset false 2 1 4 5 2	5 1 3 2 4
3 unset true unset false set true 0	2 3 1
2 unset false set true 1 2 1	-1
2 unset false set false 0	-1