JETBRAINS icpc global sponsor programming tools
HUAWEI icpc diamond multi-regional sponsor

# Problem H – Heraclosures

*Author*: Agustín Santiago Gutiérrez, Argentina

Prometheus is a programmer. He wrote a large program in Heraclosures, a programming language invented by his friend Heracles. The program contains $N$ functions, each identified uniquely by an integer from 1 to $N$. Each function is made up of many instructions, some of which are call instructions that invoke other functions.

When a call instruction in function $i$ calls a target function $j$, the full code of function $j$ is executed before continuing with the execution of function $i$. Each function may contain multiple call instructions, and it is possible for a function to call another function several times. Some functions, however, may have no call instructions at all. Following advice from his friend Sisyphus, Prometheus ensured that there are no cycles among the function calls, avoiding endless call loops.

Each function $i$ has a base execution time $B_i$, which is its execution time excluding any functions it calls. Execution times are measured in processor cycles, so $B_i$ is always a non-negative integer. The total execution time of function $i$, denoted as $T(i)$, includes both its base execution time and the total execution time of all functions it directly calls. Formally,

$$T(i) = B_i + \sum_{j \in C(i)} T(j) \ ,$$

where $B_i$ is the base execution time we already defined, and $C(i)$ is the multiset of functions that function $i$ directly calls. If function $i$ calls function $j$ multiple times, then $T(j)$ appears multiple times in this sum. If function $i$ has no call instructions, then $C(i)$ is empty, and so $T(i) = B_i$.

Prometheus wants to assess the impact of several code adjustments on the performance of his program. Tired of programming, he has hired you to implement a tool that accepts a list of events, where each event is either an update or a query.

- An update sets $B_i$ to a given value $V$. Notice that while this event may modify the base execution time of function $i$, the multiset $C(i)$ of its called functions remains the same.
- A query calculates the current total execution time $T(i)$ of function $i$, taking into account all updates that have occurred up to that point in the event list.

As a first approach, Prometheus does not want the result of each query but a single summary result. Suppose there are $q$ queries, numbered sequentially from 1 to $q$ in the list of events. Then, the summary result is calculated as:

$$\left( \sum_{k=1}^{q} k \cdot T(i_k) \right) \bmod (10^9 + 7) \ ,$$

where $T(i_k)$ is the total execution time of the function specified in the $k$-th query, taking into account all updates up to that query.

## Input

The first line contains an integer $N$ ($1 \leq N \leq 8000$) indicating the number of functions. Each function is identified by a distinct integer from 1 to $N$.

The next line contains $N$ integers $B_1, B_2, \ldots, B_N$ ($0 \leq B_i \leq 10^9$), representing that the base execution time of function $i$ is $B_i$.

The next line contains an integer $M$ ($1 \leq M \leq 8000$) denoting the number of call instructions among all the functions.

Each of the next $M$ lines describes a call instruction with two integers $F$ and $G$ ($1 \leq F, G \leq N$ and $F \neq G$), indicating that function $F$ contains a call instruction to function $G$.

The next line contains an integer $E$ ($1 \leq E \leq 10^6$) representing the number of events in the list that your tool must accept.

Each of the next $E$ lines describes an event, in the order they appear in the list.

If the event is an update, the line contains the uppercase letter "`U`" followed by two integers $I$ ($1 \leq I \leq N$) and $V$ ($0 \leq V \leq 10^9$), denoting that the base execution time of function $I$ must be set to $V$.

If the event is a query, the line contains the uppercase letter "`Q`" followed by an integer $J$ ($1 \leq J \leq N$), indicating that the total execution time of function $J$ must be determined, taking into account all previous updates that appear in the list.

It is guaranteed that there are no cycles among the function calls, and there is at least one query in the list of events.

## Output

Output a single line with an integer indicating the summary result of all the queries.

| **Sample input 1** | **Sample output 1** |
|---|---|
| 3<br>10 20 100<br>3<br>1 2<br>2 3<br>1 3<br>3<br>Q 1<br>Q 2<br>Q 3 | 770 |

**Explanation of sample 1:**
Total execution times for each query are: 230, 120 and 100.

| **Sample input 2** | **Sample output 2** |
|---|---|
| 2<br>42 10<br>2<br>2 1<br>2 1<br>5<br>Q 2<br>Q 1<br>U 2 0<br>Q 1<br>Q 2 | 640 |

**Explanation of sample 2:**
Total execution times for each query are: 94, 42, 42 and 84.