2010 Canadian Computing Competition
Day 1, Question 1

Barking Dogs!

Input: from standard input
Output: to standard output
Source file: `dogs.c`

You live in a neighbourhood of dogs. Dogs like dogs. Dogs like barking even better. But best of all, dogs like barking when other dogs bark.

Each dog has a collection of dogs that can hear him/her. Each dog has a delay time in barking if they hear another dog bark.

Dog 1 always starts barking first, and this first bark occurs during second number 0.

Your job is to figure out how many times each dog has barked in the first T seconds (inclusive). You can assume that sound travels instantly from the mouth of one dog into the ear of another.

Each dog spends any given second doing one of three things: sleeping, waiting, or barking. If dog i hears a bark during a second n when it is sleeping, the dog wakes up and waits during seconds $n+1$ through $n+w_i-1$ inclusive, barks during second $n+w_i$, then goes back to sleep from second $n+w_i+1$ onward. If a dog hears a bark during a second in which it is waiting or barking, it ignores the bark.

During second number 0, all the dogs except Dog 1 are sleeping.

Input Specification

The first line of input is D ($1 \leq D \leq 1000$), the number of dogs in the neighbourhood.

The next D lines of each contain an integer w_i ($1 \leq w_i \leq 1000$) representing the time (in seconds) that dog i waits before considering to bark upon hearing a bark.

The next line contains the number F ($1 \leq F \leq 10,000$). On each of the next F lines, there are two integers: i and j, representing that when dog i barks, dog j hears this bark. It is never the case that $i = j$.

The next line (which is the last line of input) contains the integer T ($1 \leq T \leq 1000$), the number of seconds during which your program is to monitor the dogs.

Output Specification

Produce one line of output for each dog in order from dog 1 to dog D. On line i, output the number of seconds between 1 and T inclusive that dog i spent barking.

Sample Input 1

```
3
1
1
3
3
1 2
2 3
3 1
10
```

Output for Sample Input 1

```
3
2
2
```

Sample Input 2

```
3
3
1
3
3
1 2
2 3
3 1
10
```

Output for Sample Input 2

```
2
2
1
```

2010 Canadian Computing Competition
Day 1, Question 2

Tree Pruning

Input: from standard input
Output: to standard output
Source file: `prune.c`

We are given a rooted tree with N nodes in which each node has at most two children. Each node may be black or white. We define a "prune" as the deletion of a node and the subtree rooted at that node from the tree. Given an integer D, find the minimum number of "prunes" required to obtain a tree in which the number of white nodes minus the number of black nodes is exactly D, or determine that it is impossible to do so.

Input Specification

The first line of input will contain two integers N ($1 \leq N \leq 300$) and D ($-N \leq D \leq N$), representing the number of nodes in the tree and the value of the required difference, respectively. The next N blocks of input each contain the description of a node. The first line of each block contains three integers: the id of the node (a unique integer between 0 and $N-1$), the colour of the node (1 for a white node, 0 for a black node) and an integer C that represents the number of children of the node. C lines follow, each one containing an integer that represents the id of one child. The root of the tree is the node with id 0.

Output Specification

On one line, output the minimum number of "prunes", as mentioned in the problem description. If it is impossible to obtain the required difference D, output -1.

Sample Input

```
6 3
0 1 2
1
3
1 1 2
2
5
2 1 1
4
3 1 0
4 0 0
5 1 0
```

Output for Sample Input

```
1
```

2010 Canadian Computing Competition
Day 1, Question 3
Wowow

Input: from standard input
Output: to standard output
Source file: `wowow.c`

In the World of World of Warcraft, there is a very competitive ladder system. Sometimes players will change their rating. Also, new players (including more and more of your friends!) are constantly joining the game.

You and your group of friends would like to maintain a simple database with your scores, and you, as the computer scientist of the group, have been charged with the responsability of maintaning it. Don't let your friends down!

Input Specification

The input will consist of an integer N ($1 \leq N \leq 1,000,000$), followed by N lines. Each of these N lines will correspond to one of the following three formats:

- `N` X R, where `N` is the character `'N'` to indicate a new friend has been added, X is a number ($1 \leq X \leq 1,000,000$) identifying this new friend, and R ($1 \leq R \leq 10^8$) is the rating of this new friend.
- `M` X R, where `M` is the character `'M'` to indicate a modification of an existing friend, X is a number ($1 \leq X \leq 1,000,000$) identifying one of your friends, and R is the new rating assigned to this existing friend.
- `Q` K, where `Q` is the character `'Q'` to represent a query, K is a number ($1 \leq K \leq 1,000,000$), and K is at most the number of your friends that have a rating at this point.

You may assume there will be no identical ratings in the input.

Output Specification

For each line of input of the format `Q` K, you will output a line containing the identifier of the Kth highest rated person in the database at that point. Note that when $K = 1$, that is the top rated person, and $K = 2$ is the second best rated person, and so on.

Sample Input

```
7
N 10 1000
N 3 1014
Q 1
M 10 2000
Q 1
N 65 1950
Q 2
```

Output for Sample Input

```
3
10
65
```