7774 Expect to Wait

Mayor Adam East wants to improve the public transport network of Harshel city by introducing the network of stations with unicycles. Any person who owns a special card can come to a station and request a unicycle to ride or drop one.

The procedure of requesting a unicycle is simple. The person enters a *queue*. If there is a unicycle available, then the first person from the queue takes it immediately. Otherwise, people in the queue wait until someone drops a unicycle at the station.

Let the *wait time* be the time that person spends between the request (entrance to the queue) and obtaining a unicycle. If the person does not receive a unicycle at all, then the wait time is infinity. The total wait time is the sum of wait times for each person.

Adam already knows the schedule of all the people for every day. He knows at what times people request and drop unicycles at the Central Station that can hold any number of unicycles at the same time. The only thing he does not know is how many unicycles should be placed there at the start of each day. He asks you several questions to calculate the total wait time given the starting number of unicycles.

Input

The input file contains several test cases, each of them as described below.

The first line contains n and q ($1 \leq n, q \leq 10^5$), where n is the total number of unicycle requests and unicycle drops at the Central Station, and q is the number of questions Adam asks you. The next n lines describe operations at the Central Station. Each line contains one description of operation:

- '+ t k' when k unicycles are dropped at time t;
- '- t k' when k people request unicycles at time t.

For each of the described operations $1 \leq t \leq 10^9$ and $1 \leq k \leq 10^4$. The last line of the input contains q different integers b_i ($0 \leq b_i \leq 10^9$) — the number of unicycles at the start of the day.

The operations are given in the strongly increasing order of time.

Output

For each test case, the output must follow the description below.

The output shall consist of q lines. The i-th line shall display the total wait time for the case of b_i unicycles at the start of the day. If the total wait time is infinite, then the corresponding line shall display the word 'INFINITY'.

Sample Input

```
5 4
- 1 1
- 2 2
+ 4 1
- 6 1
+ 7 2
0 3 1 2
```

Sample Output

```
INFINITY
0
8
3
```