

C. Epidemic in Monstropolis

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

There was an epidemic in Monstropolis and all monsters became sick. To recover, all monsters lined up in queue for an appointment to the only doctor in the city.

Soon, monsters became hungry and began to eat each other.

One monster can eat other monster if its weight is **strictly greater** than the weight of the monster being eaten, and they stand in the queue next to each other. Monsters eat each other instantly. There are no monsters which are being eaten at the same moment. After the monster A eats the monster B , the weight of the monster A increases by the weight of the eaten monster B . In result of such eating the length of the queue decreases by one, all monsters after the eaten one step forward so that there is no empty places in the queue again. A monster can eat several monsters one after another. Initially there were n monsters in the queue, the i -th of which had weight a_i .

For example, if weights are $[1, 2, 2, 2, 1, 2]$ (in order of queue, monsters are numbered from 1 to 6 from left to right) then some of the options are:

1. the first monster can't eat the second monster because $a_1 = 1$ is not greater than $a_2 = 2$;
2. the second monster can't eat the third monster because $a_2 = 2$ is not greater than $a_3 = 2$;
3. the second monster can't eat the fifth monster because they are not neighbors;
4. the second monster can eat the first monster, the queue will be transformed to $[3, 2, 2, 1, 2]$.

After some time, someone said a good joke and all monsters recovered. At that moment there were k ($k \leq n$) monsters in the queue, the j -th of which had weight b_j . Both sequences (a and b) contain the weights of the monsters in the order from the first to the last.

You are required to provide one of the possible orders of eating monsters which led to the current queue, or to determine that this could not happen. Assume that the doctor didn't make any appointments while monsters were eating each other.

Input

The first line contains single integer n ($1 \leq n \leq 500$) — the number of monsters in the initial queue.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$) — the initial weights of the monsters.

The third line contains single integer k ($1 \leq k \leq n$) — the number of monsters in the queue after the joke.

The fourth line contains k integers b_1, b_2, \dots, b_k ($1 \leq b_j \leq 5 \cdot 10^8$) — the weights of the monsters after the joke.

Monsters are listed in the order from the beginning of the queue to the end.

Output

In case if no actions could lead to the final queue, print "NO" (without quotes) in the only line.

Otherwise print "YES" (without quotes) in the first line. In the next $n - k$ lines print actions in the chronological order. In each line print x — the index number of the monster in the current queue which eats and, separated by space, the symbol 'L' if the monster which stays the x -th in the queue eats the monster in front of him, or 'R' if the monster which stays the x -th in the queue eats the monster behind him. After each eating the queue is enumerated again.

When one monster eats another the queue decreases. If there are several answers, print any of them.

Examples

input

6 1 2 2 2 1 2 2 5 5
output
YES 2 L 1 R 4 L 3 L

input
5 1 2 3 4 5 1 15
output
YES 5 L 4 L 3 L 2 L

input
5 1 1 1 3 3 3 2 1 6
output
NO

- Note**
- In the first example, initially there were $n = 6$ monsters, their weights are $[1, 2, 2, 2, 1, 2]$ (in order of queue from the first monster to the last monster). The final queue should be $[5, 5]$. The following sequence of eatings leads to the final queue:
- the second monster eats the monster to the left (i.e. the first monster), queue becomes $[3, 2, 2, 1, 2]$;
 - the first monster (note, it was the second on the previous step) eats the monster to the right (i.e. the second monster), queue becomes $[5, 2, 1, 2]$;
 - the fourth monster eats the mosnter to the left (i.e. the third monster), queue becomes $[5, 2, 3]$;
 - the finally, the third monster eats the monster to the left (i.e. the second monster), queue becomes $[5, 5]$.

Note that for each step the output contains numbers of the monsters in their current order in the queue.