

Spheres

[English version](#)

[Wersja polska](#)

John has a certain number of spheres. Almost all of them have identical weight apart from one. There are a lot of them and John cannot say which one differs from the other ones by himself. You can help him to determine which sphere it is by using the pair of scales.

Input/Output

In the first line of the input there is one integer n [$3 \leq n \leq 100\,000$] that stands for the number of spheres which John has. The spheres are numbered from 1 to n .

You can give John two types of orders (just print them to standard output):

- **WEIGHT** $m a_1 a_2 \dots a_m b_1 b_2 \dots b_m$

Weighting spheres. All numbers should be separated with a space and they stand for: m - number of spheres that should be put on one of the scales (there should be the same amount of spheres on both of the scales), $a_1 a_2 \dots a_m$ - identifiers of spheres that you want John to put on the left scale and $b_1 b_2 \dots b_m$ - identifiers of spheres that you want John to put on the right scale.

After conducting the weighting John will tell you about the outcome (which you will be able to read from the standard input). Possible answers are **LEFT** - spheres on the left scale are heavier, **RIGHT** - spheres on the right answer are heavier, **EQUAL** - spheres on both scales have equal weight.

After conducting the weighting John is ready right away to execute the next order.

However, you should remember that if the weighting's number is too high John can become quite bored...

- **ANSWER** k

Answering. This order is to give information that k is the identifier of the searched sphere; however if the sphere we are looking for is lighter than the other ones you should precede that with a '-' sign.

John no longer needs you after that command (your program should end).

Example

```
John: 3
You: WEIGHT 1 1 2
John: LEFT
You: WEIGHT 1 1 3
John: EQUAL
You: ANSWER -2
```

Remark: Program should clear the output buffer after printing each line. It can be done using `fflush(stdout)` command or you can set the proper type of buffering at the beginning of the execution - `setlinebuf(stdout)`.