Competitive Programming SS23

Submit until end of contest

Problem: guessds (1.0 second timelimit)

Last week Bob learned some new data structures and as an eager student implemented one of them. Can you guess which? All of the data structures have the following methods in common:

insert: Takes a value x $(0 \leq x < 100)$ and inserts it into the data structure.

empty: Returns `"yes"` if the data structure is empty, or `"no"` otherwise.

remove: Returns (and removes) a value from the data structure; the actual behaviour depends on the data structure Bob selected. This method call is only permitted if the data structure is not empty!

As a reminder, these are the data structures he learned about and how he named them (he sometimes is a bit lazy and prefers to use abbreviations):

queue: A data structure which works by the First-In-First-Out (FIFO) principle.

stack: A data structure which works by the Last-In-First-Out (LIFO) principle.

set: An ordered set of values without duplicates. In the context of this problem, removing an element means removing the smallest element.

pq: A data structure which allows quick access to the smallest value.

It is guaranteed that Bob implemented one of the above data structures. Bob is a good programmer, so there are never any bugs in his code.

Interaction This is an interactive problem. This means that your program will not first read some intput and then write some output, but instead communicates with the jury by writing queries to standard output and reading the answers from standard input.

You can perform up to 32 operations to determine what Bob implemented. To do this just print `"? <operation> <parameters>"` to `stdout`. Each of those operations has to be one of the three described above. If the operation does return a value, Bob will print it to `stdin`. If you ask too many queries Bob will print `-1` and will go away. In this case, your program should immediately exit in order to make sure that your submission receives the WRONG ANSWER verdict instead of TIMELIMIT.

As this problem is interactive, you need to flush `stdout` after each operation, because otherwise Bob won't be able to read your request.

Java: `System.out.flush()`

C++: `cout.flush()`

Python `print()` flushes per default

Output Print `"! <abbreviation>"` after you figured out which data structure Bob implemented. Your program should terminate after printing the result. If you have questions, first look at the sample interaction.

Sample interaction

```
< ? insert 42
< ? insert 66
< ? insert 1
< ? remove
> 42
< ? remove
> 66
< ? remove
> 1
< ! queue
```