



Doctors Office

Java May'19 DSA Set&Map 1 - 1 day 17:04:55

There is a long line in front of the doctor's office. Everyone is in a queue and waiting to be examined in the order they came in. However, sometimes an old lady would come in "just to ask something" and she needs to be inserted somewhere in the existing queue. Also, some patients have pre-booked appointments and need to be examined first, so there needs to be a way to find out the count of people with a given name that are in the queue.

Your task is to write a program to help the doctors office handle the queue. A sample queue is given below:

position	0	1	2	3	4	5	6
name	Peter	Niki	Nadya	Grigor	Steven	Asya	Steven

Your program should be able to process a sequence of the following commands:

- **Append [name]** – appends a person with the specified name at the end of the queue. As a result, the command prints "OK".
- **Insert [position] [name]** – inserts a person with the specified name at the specified position in the queue (position 0 is just before the first person of the queue and position N is just after the last). In case of success, as a result, the command prints "OK". In case the position is invalid, the command does nothing and prints "Error" as a result.
- **Find [name]** – finds the number of people with the specified name that are currently waiting in the queue.
- **Examine [count]** – examine the specified count of people. The examined people are removed from the front of the queue. In case of success as a result the command prints the names of the examined people in format "[name1] [name2] ..." (at a single line, separated by space, ordered as in the queue). In case the count is invalid (bigger than the number of people in the queue), the command does nothing and prints "Error" as a result.
- **End** – indicates the end of the input commands. Prints nothing and stops the program execution. This command appears only once, at the end of the input sequence of commands.

[Submit solution](#)[All submissions](#)
[Best submissions](#)

✓ **Points:** 100
(partial)
⌚ **Time limit:** 0.5s
Java: 5.0s
📜 **Memory limit:**
64M
Java: 64M
✍ **Author:**
steven

🏷 **Tags**
Hash Tables, Linear
Data Structures
⬆ **Difficulty**
Intermediate

▼ **Allowed
languages**
java

Input



Output

For each command from the input sequence print at the standard output its result as a single line.

Constraints

- All **names** will consist of letters and digits only. No spaces are allowed. The name length will be in the range [1...50]. Names are case-sensitive.
- All positions will be in the range [0 ... 1 000 000].
- The count in the **Examine** command will be in the range [1...100].
- The total number of lines in the input will be in the range [1 ... 150 000]

Sample Tests

Input

[Copy](#)

[PROBLEMS](#)[SUBMISSIONS](#)[USERS](#)[CONTESTS](#)[ABOUT](#)HELLO, **YAVORYANKOV83**.

```
Find Ina
Append Niki
Insert 0 Peter
Append Nadya
Insert 3 Grigor
Examine 5
Append Asya
Insert 4 Steven
Append Steven
Find Asya
Find Steven
Examine 3
Find Peter
Examine 4
Find Steven
Insert 1 Ina
End
```

Java May'19 DSA Set&Map 1 - 1 day 17:04:55

Output

[Copy](#)



```
0
OK
OK
OK
OK
Error
OK
OK
OK
1
2
Peter Niki Nadya
0
Grigor Steven Asya Steven
0
Error
```

Java May'19 DSA Set&Map 1 - 1 day 17:04:55

Clarifications

No clarifications have been made at this time.