Supermarket Queue

In a **supermarket** we have a very long **queue of people**. Usually the people are served in the order of their coming. When someone comes, he / she is appended at the end of the queue. When someone is served, he / she is removed from the front of the queue. Sometimes and old lady comes and the waiting people make a place for her somewhere in the queue. Because the queue could become very long (e.g. few thousand people), the supermarket management organizes a lottery and draws a random name from time to time. After each lottery draw, the management wants to know how many persons having the winning name are currently waiting in the queue.

Your task is to **write a program to help the supermarket to handle the supermarket queue**. It should hold a queue of N items numbered from 0 to N-1, where 0 is the front of the queue and N-1 is the position of the last person (end of the queue). A sample queue is given below:

position	**0**	**1**	**2**	**3**	**4**	**5**	**6**
name	Peter	Mike	Penka	Doncho	Nakov	Asya	Nakov

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 how many people with the specified name are currently waiting in the queue. As a result the command prints an integer number (**0 or more**).
- **Serve [count]** – serves the specified count of people according to their order in the queue. The served people are removed from the front of the queue. In case of success as a result the command prints the names of the served 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.

Input

The input data is given at the standard input. It consists of a sequence of commands, each at a separate line, ending by the command "End". The commands will be valid (as described in the above list), in the specified format, within the constraints given below. There is no need to check the input data explicitly.

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 "Serve" 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

```
Append Nakov
Serve 1
Find Ina
Append Mike
Insert 0 Peter
Append Penka
Insert 3 Doncho
Serve 5
Append Asya
Insert 4 Nakov
Append Nakov
Find Asya
Find Nakov
Serve 3
Find Peter
Serve 4
Find Nakov
Insert 1 Ina
End
```

Output

```
OK
Nakov
0
OK
OK
OK
OK
Error
OK
OK
OK
1
2
Peter Mike Penka
0
Doncho Nakov Asya Nakov
0
Error
```