# Sugar Cubes

You are given **all the sugar cubes** in **a sequence on a single line, separated by a space**. After that, you will receive **commands** that modify the cubes in a different way:

* **"Add {value}"** - you must add **{value}** to the end of the sequence.
* **"Remove {value}"** - you must remove the first element in the sequence with value equal to **{value}**
* **"Replace {value} {replacement}"** you must find **the first occurrence of the element** equal to **{value}** and replace its value with the **{replacement}.**
* **"Collapse {value}"** you must **remove from the sequence every element with value less** than **{value}**, if there are such elements.

When you receive command **"Mort"** you have to **print the modified sequence** and end the program.

### Input / Constraints

* On the first line – count of sugar cubes separated by spaces – integers in range

**[-1000……1000]**

* On the next lines you will receive commands untill **"Mort"** command is received.
* The commands will always be valid.

### Output

* Print a single line the array of grains separated by a space, with the modified values.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| 1 4 5 19 13 42 69 24  Add 1  Remove 4  Replace 1 26  Mort | 26 5 19 13 42 69 24 1 | The sequence – [1 4 5 19 13 42 69 24]  We start with "Add 1" so we add 1 to the end of the sequence –> [1 4 5 19 13 42 69 24 1]. The next command is "Remove 4" –> [1 5 19 13 42 69 24 1]. The next command is "Replace 1 26" – [26 5 19 13 42 69 24 1]. We read "Mort" and print the sequence. |
| 1 2 -1 0 -3 9 8 7 2  Collapse 8  Mort | 9 8 | The sequence – [1 2 -1 0 -3 9 8 7 2]  The first command is "Collapse 8" – so we remove all the elements less than 8 – [9 8]. The last one is "Mort" so we print the sequence. |

### JS Input

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| ([[1, 4, 5, 19, 13, 42, 69, 24],  "Add 1",  "Remove 4",  "Replace 1 26",  "Mort"]) | 26 5 19 13 42 69 24 1 | The sequence – [1 4 5 19 13 42 69 24]  We start with "Add 1" so we add 1 to the end of the sequence –> [1 4 5 19 13 42 69 24 1]. The next command is "Remove 4" –> [1 5 19 13 42 69 24 1]. The next command is "Replace 1 26" – [26 5 19 13 42 69 24 1]. We read "Mort" and print the sequence. |
| ([[2, -1, 0, -3, 9, 8, 7, 2],  "Collapse 8",  "Mort"]) | 9 8 | The sequence – [1 2 -1 0 -3 9 8 7 2]  The first command is "Collapse 8" – so we remove all the elements less than 8 – [9 8]. The last one is "Mort" so we print the sequence. |

*“I am the defeater of empires, the thief of years, the ultimate reality. And my horse is called Binky, it’s a nice name.”*