# Problem 2 – Grains of Sand

*You have become an apprentice! Congratulations or maybe…… regrets. Since you are working for the almighty Anthropomorphic personification – Death he is known for his work of taking care for all the world’s sand watches, he seems to deal with this job pretty well, but you as a programmer want to help him, and make his job easier.*

You are given **all the grains** of each sand watch in **a sequence on a single line, separated by spaces**. After that, you will receive **commands** that modify the grains in a different way:

**"Add {grains}"**, means that you have to add **{grains}** to the end of the sequence, grains value will always be integer.

**"Remove {grains}"**, means that you have to remove the first element in the sequence with value equal to **{grains}** starting from the beginning. If there is no such element you have to **check if the value is a valid index** for the sequence and **remove the element at that index**, however if both the **element is not valid and the value is not a valid index** you should ignore that command.

**"Replace {toReplace} {replacement}"** you have to find **the first occurrence of the element** equal to **{toReplace}** and replace it with the **{replacement}.** If element equal to **{toReplace}** doesn’t exists in the sequence you have to ignore this command.

**"Increase {grains}"** you have to find the first element with **value** **not less than {grains}** and **increase the value of all elements** in the sequence with this value. **If no such element exists in the set, you have to take the last element from the sequence** and then **increase the value of all elements** in the sequence with this value.

**"Collapse {grains}"** you have to **remove from the sequence every element with value less** than **{grains}**, 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 sands in each watch separated by spaces – integers in range

**[-2,147,483,648……2,147,483,647]**

* 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 spaces, with the modified values.
* Allowed working **time** / **memory**: **100ms** / **16MB**

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| 1 4 5 19 13 42 69 24  Add 1  Remove 3  Remove 4  Remove 15  Replace 0 26  Replace 1 26  Mort | 26 5 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 and it looks like – [1 4 5 19 13 42 69 24 1]. The next command is "Remove 3" – [1 4 5 13 42 69 24 1]. The next command is "Remove 4" – [1 5 13 42 69 24 1]. The next command is "Remove 15" – [1 5 13 42 69 24 1]. The next command is "Replace 0 26" – [1 5 13 42 69 24 1]. The next command is "Replace 1 26" – [26 5 13 42 69 24 1]. We read "Mort", and print the sequence. |
| 1 2 -1 0 -3 9 8 7 2  Increase 10  Increase 90  Collapse 8  Mort | 8 15 14 13 8 | The sequence – [1 2 -1 0 -3 9 8 7 2]  The first one is "Increase 10" so we increase the sequence by 2 – [3 4 1 2 -1 11 10 9 4]. The next one is "Increase 90" so we increase the sequence with 4 – [7 8 5 6 3 15 14 13 8]. The next one is "Collapse 8" – so we remove all the elements less than 8 – [8 15 14 13 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.”*