Advanced C#– Debugging

The goal of this lab is to practice **debugging techniques** in scenarios where a piece of code does not work correctly. Your task is to pinpoint the bug and fix it (without rewriting the entire code).

Problem 5. Array Test

You are given a number **n** representing the size of an array of integers and on the next line the elements of the array separated by whitespace. Then, you are given an arbitrary number of commands in the format: **"[action] [i-th element] [value]"**. The commands end when you receive the string **"stop".**

The action can be **"multiply", "add", "subtract", "rshift" or "lshift"**. The actions **"multiply", "add"** and **"subtract"** have parameters. The first parameter is the number of the element that needs to be changed. The second parameter is the value with which we manipulate the element. The command **"lshift"** iterates through the array changing each element's position with 1 to the left. The first element which should go outside the array will eventually become last. E.g. {1, 2, 3} "lshift" will become {2, 3, 1}. The command **"rshift"** does the same thing but changes the positions with 1 to the right. The last element which should go outside the array, becomes first. E.g. {1, 2, 3} "rshift" will become {3, 1, 2}.

Example:

```
5
3 0 9 333 11
add 2 2
subtract 1 1
multiply 3 3
stop
```

We shift every 2nd element to the right twice. After the shifting we obtain the array **{2 2 27 333 11}**.

Output

For each action print the array's elements on a new line on the console.

Constraints

- The **number n** will be an integer in the range [1 … 15].
- Each **element of the array** will be an integer in the range [0 … 2^{63}-1].
- The **number i** and the **number of commands** will be integers in the range [1 … 10].
- The **number value** will be an integer in the range [-100 … 100]. If the command is "rshift" or "lshift" there are no parameters.

Tests

Input	Program Output	Expected Output

5
3 0 9 333 11
add 2 2
subtract 1 1
multiply 3 3
stop
	3 0 9 333 11
	3 0 9 333 11
		3 2 9 333 11
		2 2 9 333 11
		2 2 27 333 11