## Problem 2. Array Modifier

You are given **an array with integers**. Write a program to **modify the array elements** after **processing a sequence of commands** “**swap**”, “**multiply**” or “**decrease**”. The commands are as follows:

* “swap {index1} {index2}” takes **two elements** and **swaps them**.
* “multiply {index1} {index2}” takes **element at the 1st index** and **multiplies** it with the element at **2nd index**. Save the **product at the 1st index**.
* “decrease” **decreases** **all elements** in the array **with 1**.

### Input

On the **first input line** you will be given **the initial array values** separated by a single space.

On the **next lines** you will receive commands **until** you receive the **command “end”**. The **commands are** as follow:

* “swap {index1} {index2}”
* “multiply {index1} {index2}”
* “decrease”

### Output

**The output** should be printed on the console and consist **element** **of the** **modified array** – separated by “, “ (comma and single space).

### Constraints

* Commands will be: “**swap**”, “**multiply**”, “**decrease**” and “**end**”.
* **Elements of the array** will be **integer numbers** in the range **[-231**...**231]**.
* **Count of the array elements** will be in the range **[2**...**100]**.
* **Indexes** **will be always in the range of the array**.

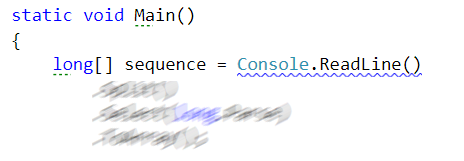
### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 23 -2 321 87 42 90 -123  swap 1 3  swap 3 6  swap 1 0  multiply 1 2  multiply 2 1  decrease  end | 86, 7382, 2369942, -124, 41, 89, -3 | 23 -2 321 87 42 90 -123 – initial values  swap 1(-2) and 3(87) ▼  23 87 321 -2 42 90 -123  swap 3(-2) and 6(-123) ▼  23 87 321 -123 42 90 -2  swap 1(87) and 0(23) ▼  87 23 321 -123 42 90 -2  multiply 1(23) 2(321) = 7383 ▼  87 7383 321 -123 42 290 -2  multiply 2(321) 1(7383) = 2369943 ▼  87 7383 2369943 -123 42 90 -2  decrease – all - 1 ▼  86 7383 2369942 -124 41 89 -3 |

# Solution

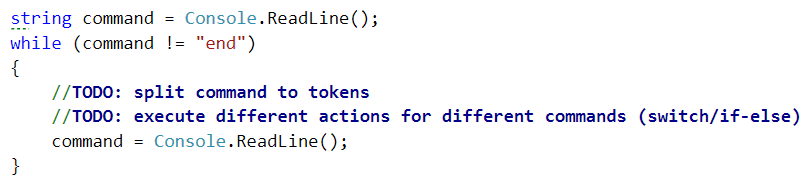
## Read Input

We receive the integer numbers in a single line separated by space. So we need to split that line by space and to parse every number. Since we are going to multiply integers we can easily overflow it so we will use long data type.

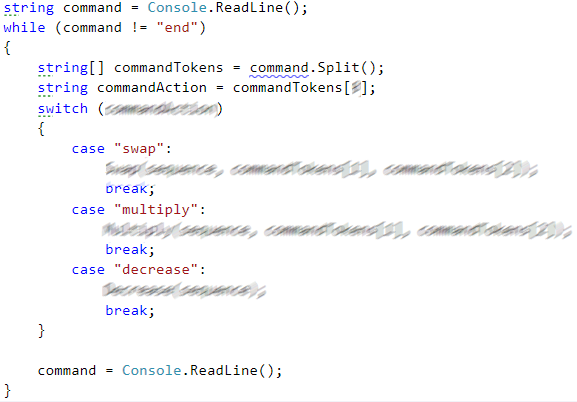


## Command Execution Loop

Now we need to read and execute the commands on the next lines until we receive “end” as command

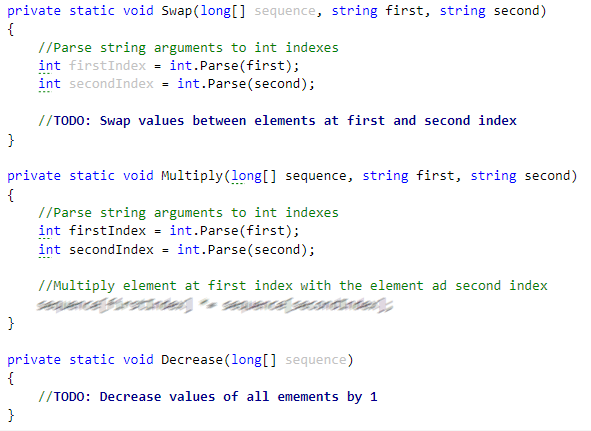


### Hint



## Helper Methods

We could define 3 methods to make the code more readable. And every method executes the appropriate command from the input. The methods receive as parameters the sequence and the parameters from the received command



## Print Result

Finally we need to print the final state of the sequence separated by a comma and space.

code4