




Problem of the day. Next problem in **12h 27m 10s**

You will get  **5–10 hypercoins** for solving it.

Java → Implementation of basic algorithms → Dynamic array in Java

Java → Allocated memory

 Hard  36 minutes 

5 users solved this problem. Latest completion was **3 days ago**.

You've postponed this problem 4 days ago.

Maybe now is a good time to try it out again?

Code Challenge — Write a program

Assume that in addition to the scaling factor, there is a factor responsible for deallocating unnecessary memory after removing elements from the list. You will be given two factors: one for increasing and one for reducing the allocated memory. Following that will be queries for abstract addition and removal of clusters of elements.

Your task is to return the allocated memory for each 'count' query at that moment. Multiplication on a fraction is rounded up, division by a fraction is rounded down. The initially allocated memory is 2.

Input: In the first line there's a number of queries and both the scaling (≥ 1.1) and downscaling (≥ 1.1) factors. Following that are queries of these types:

- add count;
- remove count;
- count.

Output: For every 'count' query, write down the amount of allocated memory at the moment.

Sample Input 1:

```
3 2.0 2.0
add 100
delete 50
count
```

Sample Output 1:

```
64
```

[Code Editor](#)

[IDE](#)



✓ **IDE is responding** IntelliJ IDEA 2019.3

✓ **Plugin is responding** 3.2-2019.3-3686

Wrong, but keep on trying and never give up! 1 code style error

Failed test #2 of 5. Wrong answer

What is a test and why don't we always show it?

[Download the failed test #2 for free](#)

One test is free once a day, more tests for hypercoins.

Time limit: 8 seconds Memory limit: 256 MB

[Comments \(2\)](#)

[Hints \(1\)](#)

[Useful links \(0\)](#)

[Solutions \(2\)](#)

SK

Share something, Sergey Kubatko

Post

Please do not post solutions here

KP

Konstantin Proskurnya [14 days ago](#) [Report](#)

cast to int only in count section;
use only math.floor, not math.ceil in both cases and use it not inside while loop, but after.

 2 [Reply](#)