# Exam Preparation

Submit your solutions here: <https://alpha.judge.softuni.org/contests/exam-preparation/2943>

## Below Average Only

Given an array of integer numbers and its size, you will have to create a program for certain calculations. Your task is to find the average number for the given array then only get the numbers *below or equal to* that average.

Create two sums from these numbers: a sum of the elements on even positions and a sum of the elements on odd positions. Then multiply those sums and print the final product on the console.

### Input

### On the first line you will receive an integer number – the length of the array.

* On the **second line** you will receive a **sequence of integer numbers** – the array.

### Output

### Print the expected number.

### Examples

|  |  |  |
| --- | --- | --- |
| Input | Output | Comments |
| 7  1 1 1 1 1 1 1 | 12 | The average number of the array is:  (1 + 1 + 1 + 1 + 1 + 1 + 1) / 7 = 7 / 7 = 1  *All numbers are below or equal to the average of the array.*  Sum of elements on even positions (position 0, 2, 4, 6): 4  Sum of elements on odd positions (position 1, 3, 5): 3  Product: 4 \* 3 = 12 |
| 8  4 2 1 1 3 15 6 0 | 24 | (4 + 2 + 1 + 1 + 3 + 15 + 6 + 0) / 8 = 4  4 2 1 1 3 15 6 0  Even positions: 8, Odd Positions: 3  Product: 8 \* 3 = 24 |
| 2  1 1 | 1 |  |

## Special Items

### You are playing a game in which all symbols of a string are considered special items. Your goal is to print a sequence of different symbols and white spaces (write sentences) with no consecutive duplicates of the items unless they are "magical". White spaces should also be removed.

### Magical items are all the vowels: 'a', 'e', 'i', 'o', 'u'.

Your task is to write a program that removes the unnecessary symbols and print the expected output.

### Input

### On a single line you will receive a string.

### Output

### Print the new string on the console.

### Examples

|  |  |
| --- | --- |
| Input | Output |
| C++ is the beeest!! | C+ is the beeest! |
| I lllovve Programming! | I love Programing! |
| ttttttt | t |

## Machine Commands

You are tasked with implementing software for a computational machine, which does basic operations with integer numbers. The machine has memory in the form of a sequence of numbers. When it does an operation, it takes (removes) numbers from the end of the sequence, calculates the result using those numbers and puts it back at the end of the sequence. There are also some operations which **only add at the end or only remove from the end of the sequence**.

List of operation command:

* **{number}** – Inserting a number: **number** is a single integer number (can be positive, negative or 0). Inserts a number **at the end of the sequence** (appends to the end of the sequence).
* sum - Removes the last two numbers in the sequence, calculates their sum and adds it back to the end of the sequence.
* subtract - Removes the last element from the sequence **(a)** then removes the next last element **(b)**. Subtracts the two **(**a – b) and adds the result to the end of the sequence.   
  ***Example***: If the sequence is (1, 4, 7), the command subtract will remove 7 and 4, calculate 7 - 4 and add it back to the sequence. The result sequence will be (1, 3).
* concat - Concatenates the last two elements in the sequence (as if they were strings), **in the order they were added to it**, evaluates the result to an integer and adds it to the end of the sequence.  
  ***Example:*** If the sequence is (1, 4, 7), then concat will remove 7 and 4, concatenate 4 and 7, resulting in 47, and add it to the sequence. The result sequence will be (1, 47).
* discard - Removes the last element from the sequence.
* end – End of program.

Write a program which executes the operations described above and prints the final sequence of numbers in the order in which they were added.

### Input

Two or more lines, each indicating an operation to be done with the machine (note that a line containing a single integer is the **{number}** operation). The final line will not contain numbers and will only contain the string "end"

### Output

One or more lines, each containing a single integer, representing the numbers in the final sequence.

### Restrictions

There will be no more than 50 lines of operations in the input. The **concat** operation will never be done when the last element in the sequence is negative. All operations will be valid – there will always be enough numbers in the sequence for an operation to be executed correctly. There will always be at least 1 number in the final sequence.

### Examples

|  |  |  |
| --- | --- | --- |
| Input | Output | Comments |
| 1  4  sum  end | 5 | Inserting: 1 => 1  Inserting: 4 => 1 4  Summing the last two numbers: 4 + 1 = 5 => 5  End of program. The result output is: 5 |
| 1  4  subtract  7  end | 3  7 | Inserting: 1 => 1  Inserting: 4 => 1 4  Subtracting the last two numbers: 4 – 1 = 3 => 3  Inserting: 7 => 3 7  End of program. The result output is:  3  7 |
| 3  10  1  -1  sum  concat  sum  end | 103 | Inserting: 3 => 3  Inserting: 10 => 3 10  Inserting: 1 => 3 10 1  Inserting: -1 => 3 10 1 -1  Summing the last two numbers: -1 + 1 = 0 => 3 10 0  Contatinating the last two numbers: 100 => 3 100  Summing the last two numbers: 100 + 3 => 103  End of program. The result output is: 103 |