

Problem 2 – Letters change Numbers

Nakov likes Math. But he also likes the English alphabet a lot. He invented a game with numbers and letters from the **English** alphabet. The game was simple. You get a string consisting of a **number between two letters**. Depending on whether the letter was in front of the number or after it you would perform different mathematical operations on the number to achieve the result.

First you start with the letter **before** the number. If it's **Uppercase** you **divide** the number by the letter's **position** in the alphabet. If it's **lowercase** you **multiply** the number with the letter's position. **Then** you move to the **letter after** the number. If it's **Uppercase** you **subtract** its position from the resulted number. If it's **lowercase** you **add** its position to the resulted number. But the game became too easy for Nakov really quick. He decided to complicate it a bit by doing the same but with **multiple** strings keeping track of only the **total sum** of all results. Once he started to solve this with more strings and bigger numbers it became quite hard to do it only in his mind. So he kindly asks you to write a program that **calculates the sum of all numbers after the operations on each number have been done**.

For example, you are given the sequence "A12b s17G". We have two strings – "A12b" and "s17G". We do the operations on each and sum them. We start with the letter before the number on the first string. **A is Uppercase** and its position in the alphabet is **1**. So we divide the number 12 with the position 1 ($12/1 = 12$). Then we move to the letter after the number. **b is lowercase** and its position is 2. So we add 2 to the resulted number ($12+2=14$). Similarly for the second string **s is lowercase** and its position is 19 so we multiply it with the number ($17*19 = 323$). Then we have Uppercase G with position 7, so we subtract it from the resulted number ($323 - 7 = 316$). Finally we sum the 2 results and we get $14 + 316=330$;

Input

The input comes from the console as a **single line, holding the sequence of strings**. Strings are separated by **one or more white spaces**.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

Print at the console a single number: the **total sum of all processed numbers** rounded up to **two digits** after the decimal separator.

Constraints

- The **count** of the strings will be in the range **[1...10]**.
- The numbers between the letters will be integers in range **[1...2,147,483,647]**.
- Time limit: 0.3 sec. Memory limit: 16 MB.

Examples

Input	Output	Comment
A12b s17G	330.00	$12/1=12$, $12+2=14$, $17*19=323$, $323-7=316$, $14+316=330$
P34562Z q2576f H456z	46015.13	
a1A	0.00	