



## Unloaded Die

Consider a standard six-sided die, with sides labelled 1 through 6. We consider a die to be *fair* if each of its sides is equally likely to be landed on after rolling it. We consider a die *loaded* if its fairness is compromised. For example, if the side marked 6 twice as likely to come up as any other side, we are dealing with a *loaded* die.

For any die, define the expected value of rolling the die to be equal to the average of the values of the sides weighted by the probability of those sides coming up. Intuitively, this is the number you would get if you rolled the die many times and averaged all the results together. A *fair* die has an expected result of 3.5. That is, since all sides are weighed the same, they each have probability of  $1/6$ , and we get:

$$1/6 + 2/6 + 3/6 + 4/6 + 5/6 + 6/6 = 3.5$$

Suppose you are given a *loaded* die, and you would like to "unload" it to make it more closely resemble a *fair* die. To do so, you can erase one side's label and replace it with a new (real) number. You want to do so in such a way that

- 1) The expected result of rolling the die is 3.5, just like a *fair* die, and
- 2) The absolute value of difference between the old label and the new label on the side you change is as small as possible.

### Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will consist of a single line with six decimal numbers, where the  $i^{\text{th}}$  number ( $i = 1..6$ ) is the probability that the side with value  $i$  is rolled. All of these numbers will be between **0.0** and **1.0**, and they are guaranteed to sum to **1.0**.

### Output

Output a single number on a single line: the absolute value of the difference between the label you erase and the label you write in. Output this number to exactly 3 decimal places, rounded.

### Sample Input

### Sample Output

0.166666 0.166667 0.166667 0.166666 0.166667 0.166667	0.000
0.2 0.2 0.1 0.2 0.2 0.1	1.000