Problem 1. Compute Mean and Variance (30 pts)

Calculating the mean and variance is an important operation in mathematics, and we want you to implement the functionality to compute the mean and variance of a series of numbers in RISC-V code.

- Assume that there are a total of k numbers as input.
- The formula for calculating the mean is given by $x_{mean} = \frac{1}{k}\sum_{i=1}^{k} x_i$, where x_i represents the i-th input number.
- The formula for calculating the variance is $\frac{1}{k}\sum_{i=1}^{k}(x_i - x_{mean})^2$,

Input Format

The input consists of several lines.

- The first line is a positive integer k $(1 \leq N \leq 100)$, representing the dimension of the vector.
- The next k lines contain the elements of the first vector, with each line containing a floating-point number.

For this problem, you need to use the **double** data type in RV32 to perform the calculations. Therefore, when reading the input floating-point numbers, use system call NO. 7.

Output Format

The output consists of **2** lines.

- The first line contains a double floating-point number representing the mean. Use system call NO. 3 to output it.
- The second line contains a double floating-point number representing the variance. Use system call NO. 3 to output it.

You can use the following code to implement a newline:

```
li a7, 11                # System call 11: print character
li a0, 10                # ASCII code 10 corresponds to newline '\n'
ecall
```

Samples

Sample 1

Input

```
3
1
2
3
```

Output

```
2.0
0.6666666666666666
```

Sample 2

Input

```
3
0
2
4
```

Output

```
2.0
2.6666666666666665
```

Problem 2. Binary Search Method for Calculating Square Roots (40 pts)

The binary search method is a common technique for calculating the square root of a number. In this problem, we want you to use this method to find the square root of a positive number.

To find the square root of a positive number a, it can be equivalent to finding the root of the equation $f(x) = x^2 - a$. The overall calculation process using the binary search method is as follows:

1. We will first provide you with two starting points x_1 and x_2 for the binary search algorithm. These two points define the range where the root of the equation is located.
2. Next, you need to iteratively approximate the root within this numerical interval using the binary search method.
3. In each iteration, you will calculate $x_3 = \frac{x_1+x_2}{2}$. If $|f(x_3)| < 1e-6$, we consider that you have found the root of the equation and can exit the iteration process.
4. If $|f(x_3)| \geq 1e-6$, you need to continue updating the interval.
5. The rule for updating the interval is as follows: if $f(x_3) \times f(x_1) < 0$, set $x_2 = x_3$; otherwise, set $x_1 = x_3$.

Input Format

The input consists of 3 lines, each containing a floating-point number.

- The first line represents a, the positive number whose square root needs to be calculated, with the guarantee that $a > 0$.
- The second and third lines represent the starting points x_1 and x_2 for the binary search algorithm, with the guarantee that $x_1 < x_2$, $f(x_1) \neq 0$, $f(x_2) \neq 0$, and $f(x_1) \times f(x_2) < 0$.

For this problem, you need to use the **double** data type in RV32 to perform the calculations. Therefore, when reading the input floating-point numbers, use system call NO. 7.

Output Format

A double floating-point number x_0 representing the root of the equation, satisfying the condition $|f(x_0)| < 1e-6$. Use system call NO. 3 to output it.

Samples

Sample 1

Input

```
25
0
25
```

Output

```
5.000000074505806
```

Sample 2

Input

```
36
0
10
```

Output

```
5.999999940395355
```

Problem 3. Newton's Method for Calculating Square Roots (30 pts)

Newton's method is another common technique for calculating the square root of a number. In this problem, we want you to use Newton's method to find the square root of a positive number a . The overall calculation process is as follows:

1. We will first provide you with an initial guess $x_{current} = x_0$ and the positive number a for which the square root needs to be calculated.
2. Next, you need to calculate the new (x) value using the iterative formula $x_{\text{next}} = \frac{1}{2}(x_{\text{current}} + \frac{a}{x_{\text{current}}})$.
3. If $|x_{\text{next}} - x_{\text{current}}| < 1e-6$, we consider that you have found the square root of a and can exit the iteration process.
4. If $|x_{\text{next}} - x_{\text{current}}| \geq 1e-6$, you need to update $x_{\text{current}} = x_{\text{next}}$ and continue iterating.

Input Format

The input consists of 2 lines, each containing a floating-point number.

- The first line represents a, the positive number whose square root needs to be calculated, with the guarantee that $a > 0$.
- The second line represents the guess x_0, with the guarantee that $x_0 > 0$.

For this problem, you need to use the **double** data type in RV32 to perform the calculations. Therefore, when reading the input floating-point numbers, use system call NO. 7.

Output Format

A double floating-point number x_0 representing the root of the equation. Use system call NO. 3 to output it.

Samples

Sample 1

Input

```
16
0.5
```

output

```
4.000000000000085
```

Sample 2

Input

```
20
1
```

output

```
4.47213595500161
```