

Babylonian Square Root Algorithm

Problem ID: a05p03babylonian

This exercise is based on chapter 3.5 in the textbook.

Let us implement the Babylonian square root algorithm. Here is a pseudocode description of the algorithm:

- Let n be the number for which to find the square root.
- Let g be the initial guess.
- Let t be the tolerance.
- Let g' be our previous guess, initialized to 0.
- While the absolute difference between g and g' is greater than t :
 - $g' = g$.
 - $g = \frac{\frac{n}{g} + g}{2}$, the average of $\frac{n}{g}$ and g .

Input

Input consists of three lines and they are as follows:

- n - an integer for which to find the square root, where $1 \leq n \leq 100\,000$.
- g - an integer representing from where to start the search for the actual square root, where $1 \leq g \leq 100\,000$.
- t - a floating point number, indicating the minimum change between iterations until the algorithm stops, where $0.1 \leq t \leq 0.000\,01$. The number is given with at most 5 digits after the decimal point.

Output

Output consists of two lines. The first line contains the square root of n , and the second line contains the number of repetitions until the tolerance was reached. The output of the square root should have an absolute or relative error of at most 10^{-4} .

Sample Input 1

```
2
1
0.0001
```

Sample Output 1

```
1.4142
4
```

Sample Input 2

```
9
2
0.00001
```

Sample Output 2

```
3.0000
5
```

Sample Input 3

```
1
1
0.00001
```

Sample Output 3

```
1.0000
1
```