Newton's Method Calculation

A quadratic equation with one variable (一元二次方程) is a equation that can be rearranged in standard form as $ax^2 + bx + c = 0$, where a ($a \neq 0$), b, and c are constants.

The Newton's method is an iterative method that can be used to find the roots of a function. It involves choosing an initial guess and then repeatedly refining it until the solution is found.

In this problem, we are going to use the Newton' method to solve a quadratic equation with one variable. The inputs correspond to values **a, b, and c respectively** and the output are the solutions x_1 and x_2 in **increasing order**.

Hint:

1. All the numbers in the calculation are taken as **double-precision**.
2. We make sure there must be **two different solutions** to this equation.
3. We make sure a, b and c are **non-zero**.
4. When you substitute your solution x_n into the equation, the absolute value of the result $|f(x_n)|$ should be less than **1e-6**. Once a result satisfying to this precision, it is considered to be a correct result, and iteration cannot be continued.
5. You can use `abs.d` to get the absolute value.
6. About the output, there is only **one newline** between the smaller value and the bigger value.

The process is like this:

After reading a, b and c, first calculate $-\frac{b}{2a}$, then minus 1e-6 as the first guess $x_{fSmaller}$ for smaller x_n , and add 1e-6 as the first guess $x_{fBigger}$ for bigger x_n . Let's calculate smaller x_n as an example. The euation is like $f(x) = ax^2 + bx + c$,and calculate $|f(x_n)|$ with our first guess $x_{fSmaller}$, and check whether $|f(x_n)|$ is less than **1e-6**. If it satisfies, then this x_n is one of the results, but if not, calculate x_{n+1} like this $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{ax_n^2 + bx_n + c}{2ax_n + b}$ as the next iteration's input.

Standard:

To make sure the **precision**, please use data like this:

```
.data
    precision: .double 0.000001
    newline: .asciiz "\n"
```

And in text: You can use `ldc1 $f20, precision` for loading the precision, when calculating whether the result is less than the precision, you can use `c.lt.d $f8, $f20`.

When reading a, b and c it can be like this:

```
.text
    li $v0, 7
    syscall
    mov.d $f14, $f0
    syscall
    mov.d $f16, $f0
    syscall
    mov.d $f18, $f0
```

Sample inputs and outputs:

Sample Input1:

```
1
4
1
```

Sample Output1:

```
-3.732050820939485
-0.2679491790605153
```

Sample Input2:

```
1
-5
6
```

Sample Output2:

```
1.99999994826938
3.000000051730621
```

Sample Input3:

```
1.5
9.3
-4.7
```

Sample Output3:

```
-6.669780621053905
0.46978062105390345
```

Sample Input4:

```
5.2232
17.3754
3.2523
```

Sample Output4:

```
-3.127487307501963
-0.19909409853265192
```