Homework Assignment

This assignment has to be made individually

4 Interpol

You are going to find a root of a quadratic function f by a numeric approach. Provided on input are the coefficients defining f as well as an interval that should contain the root. This interval is specified by two double values x_1 and x_2 between which the f changes sign, i.e., $f(x_1) > 0$ and $f(x_2) < 0$ or the other way around. Your program should find the root by repeatedly halving the interval. Take the middle m between the endpoints and continue with this point and one of the two endpoints, keeping the property invariant that f changes sign in the interval. Stop when the interval is smaller than a small value. For this exercise, this value is 0.000001. Make it a constant in your program.

An advanced version calculates the root for an arbitrary polynomial.

4.1 Input

1. Three decimal numbers (size of `double`) specifying the coefficients a, b, and c of the quadratic function $f(x) = ax^2 + bx + c$. You may assume that $|a| > 0.0001$
2. The word `in` .
3. Two decimal numbers x_1 and x_2 specifying the begin and end point of the initial interval. It is not guaranteed that f changes sign in this interval.

In the advanced version, any positive number of decimal numbers is allowed before the word `in`. These are interpreted as the coefficients of a polynomial, highest exponent first. E,g,. input `2 3 4 5` represents the polynomial $2x^3 + 3x^2 + 4x + 5$.

Challenge: find an efficient algorithm for the calculation of the value of the polynomial.

4.2 Output

Either `root:` followed by the root rounded off to 3 decimals (see below) or

`unusable interval:` followed by the values of $f(x_1)$ and $f(x_2)$, rounded off to 3 decimals. This in case f doesn't change sign between x_1 and x_2.

4.3 Remarks

4.3.1 Design

Make a well-structured program, with properly designed functions and methods.

Provide at least the following functions.

- `f` is the function that has to be analysed, it computes $ax^2 + bx + c$; make your own decision on the parameters and return type.
- `double rnd(double d)` rounds off d to three decimals. Use this function when printing doubles (and use it in printing only!). You could make use the function `double Math.round(double x)` that rounds off x to the nearest whole number.

4.3.2 Comma f*ing

Java (in particular, the Scanner object) consults the *locale* setting of your computer to see whether it should expect a decimal comma or a decimal point when inputting numbers. For the tests that Peach performs, this setting is irrelevant, since Peach uses its own environment and tests with corresponding input (it uses decimal points).

If you nevertheless really want to override this behaviour, add the line

```
Locale.setDefault( Locale.US );
```

to the beginning of your programming, i.e., the beginning of the body of `main`. (Make sure you have `import java.util.*;` before the class definition.) The `print` and `println` methods seem not to use this setting and always produce decimal points.

Do not panic when your results do not exactly match Peach's expectations. Floating point arithmetic has some tolerances and two correct algorithms might give different results. Make sure that you round off at printing and not earlier.

4.4 Example

4.4.1 Input

```
 1.23 4.5 -6.789
 in  0    2
```

4.4.2 Output

```
  root: 1.148
```

4.5 Example

4.5.1 Input

```
  1.23 4.5 -6.789
in  -5.0   5.0
```

4.5.2 Output

```
  unusable interval: 1.461, 46.461
```

4.6 Example

4.6.1 Input

```
  1 0 0 0
in  -5.0   5.0
```

4.6.2 Output

```
  root: 0.0
```