Question 4:

a) The currently is using the relative error as its stopping condition
b) Calculation the square root
    a. Necessary changes:
        - Add class Result to print out the result of square root and loops of progress
        - Minimal changes:
            a. Add loops into while-loop to count
            b. Add optional to use Relative Error to compare the results of between Absolute relative error and amount of error.
            c. Add zero checking just for safety check
        -
    b. Result:

```
sqrt(0.25) ≈ 0.500 (loops=1)
sqrt(5000000000000) ≈ 2236339.332 (loops=32)
sqrt(5000000) ≈ 2236.174 (loops=22)
sqrt(5) ≈ 2.236 (loops=11)
sqrt(0.005) ≈ 0.071 (loops=14)
sqrt(0.000005) ≈ 0.002 (loops=19)

Using AMOUNT OF ERROR (interval width) ≤ 0.001
sqrt(0.25) ≈ 0.500 (loops=1)
sqrt(5000000000000) ≈ 2236067.978 (loops=53)
sqrt(5000000) ≈ 2236.068 (loops=33)
sqrt(5) ≈ 2.235 (loops=12)
sqrt(0.005) ≈ 0.071 (loops=10)
sqrt(0.000005) ≈ 0.003 (loops=10)
```

c) Have been changed:
    a. Comparison of result:

```
sqrt(0.25) ≈ 0.500 (loops=1)
sqrt(5000000000000) ≈ 2236339.332 (loops=32)
sqrt(5000000) ≈ 2236.174 (loops=22)
sqrt(5) ≈ 2.236 (loops=11)
sqrt(0.005) ≈ 0.071 (loops=14)
sqrt(0.000005) ≈ 0.002 (loops=19)

Using AMOUNT OF ERROR (interval width) ≤ 0.001
sqrt(0.25) ≈ 0.500 (loops=1)
sqrt(5000000000000) ≈ 2236067.978 (loops=53)
sqrt(5000000) ≈ 2236.068 (loops=33)
sqrt(5) ≈ 2.235 (loops=12)
sqrt(0.005) ≈ 0.071 (loops=10)
sqrt(0.000005) ≈ 0.003 (loops=10)
```

d) Table of results of the sqrt method

Value	Amount of error		Relative error	
	Square root	Loops	Square root	Loops
0.25	0.5013	7	0.499973	9
5000000000000	2236067.978	53	2236339.332	32
5000000	2236.068	33	2236.174	22
5	2.235	12	2.236	11
0.005	0.071	10	0.071	14
0.000005	0.003	10	**0.002**	19

e) From observing the result, using amount of error is faster than using relative error when squaring root of the small number (less than 1). However, the result is not ensured to correct like relative error

f) According to table, handling the large number is that using amount of error take more accurate even through it take more couple loop than relative error.

g)

```
=== Precision 0.001 ===
Value            | RelError Result           | Loops  | AbsError Result           | Loops
-----------------------------------------------------------------------------------------------
0.25             | 0.500                     | 1      | 0.500                     | 1
5000000000000    | 2236339.332               | 32     | 2236067.978               | 53
5000000          | 2236.174                  | 22     | 2236.068                  | 33
5                | 2.236                     | 11     | 2.235                     | 12
0.005            | 0.071                     | 14     | 0.071                     | 10
0.000005         | 0.002                     | 19     | 0.003                     | 10

=== Precision 0.000001 ===
Value            | RelError Result           | Loops  | AbsError Result           | Loops
-----------------------------------------------------------------------------------------------
0.25             | 0.500000                  | 1      | 0.500000                  | 1
5000000000000    | 2236067.620753            | 42     | 2236067.977500            | 63
5000000          | 2236.067794               | 32     | 2236.067978               | 43
5                | 2.236067                  | 21     | 2.236068                  | 22
0.005            | 0.070711                  | 24     | 0.070710                  | 20
0.000005         | 0.002236                  | 29     | 0.002236                  | 20
```

- According to both analyzing table and table of results, we can conclude that using amount of error take more advantage in calculating for large numbers than using relative error and reverse with calculating the small numbers, relative error show its benefit in aspect of accurate. However, we can see that more loops or more time running maybe can get more accurate result.

Question 5:

a. *[2 marks]* Consider the following function *p*(*x*). What is the maximum number of zeros for this function? How can you tell?

$$p(x) = 8x^4 - \frac{89}{9}x^2 - 1$$

Because this is a biquadratic equation, so it has maximum 4 number of zeros.

The solution:

➔ Place substitute: $t = x^2$ with $t \geq 0$

➔ So the function will be $8t^2 - \frac{89}{9}t - 1$

➔ Each t has 2 roots: $x = \pm\sqrt{t}$

b. *[2 marks]* In order for the method of bisection to work, you need two starting values – one where the value of the function is positive and one where the value of the function is negative. Find a set of starting values for as many zeros of *p*(*x*) as you can identify. List the starting points.

- *Calculating function:*

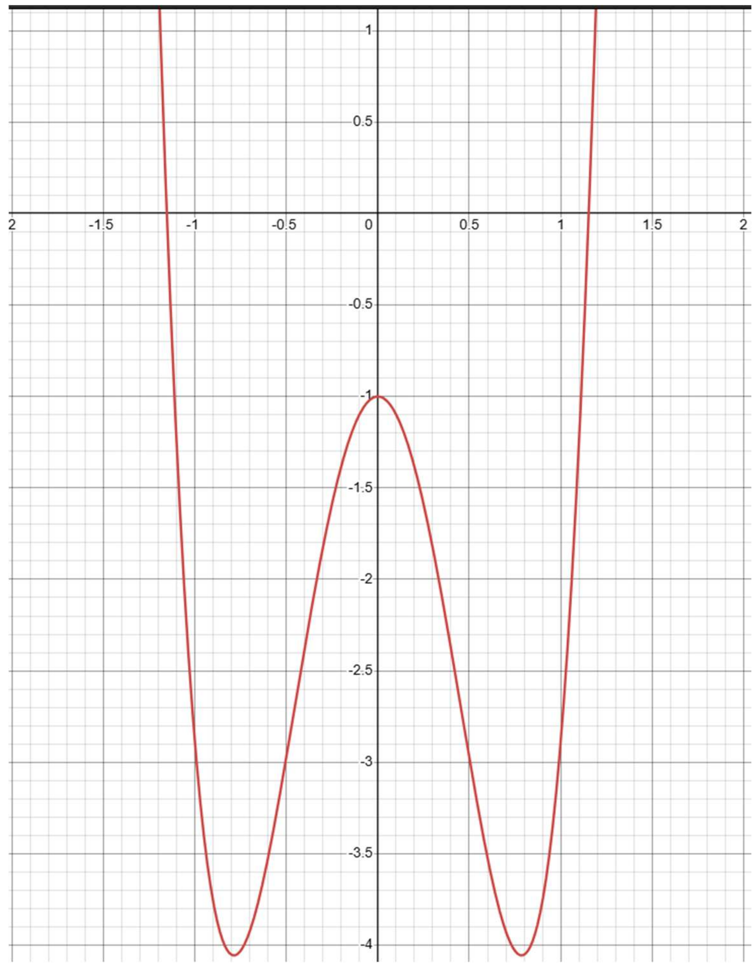
➔ *Take function from a:* $8t^2 - \frac{89}{9}t - 1$

➔ $\Delta = \left(\frac{89}{9}\right)^2 - 4 * 8 * (-1) \approx 129.79$

➔ $t_1 \approx \frac{-\frac{89}{9}+\sqrt{129.79}}{2*8} \approx 1.33$ $\quad t_2 \approx \frac{-\frac{89}{9}-\sqrt{129.79}}{2*8} \approx -1.33$

From 2 values from t_1 and t_2, we can get 2 real values and 2 complex values

$$x \approx \pm\sqrt{1.33} \approx \pm 1.153$$



Source: Desmos | Graphing Calculator

- For the negative zero, a suitable start interval is [-1.2,-1.0]
- For the positive zero, a suitable start interval is [1.0,1.2]