ECE 122: Introduction to Programming for ECE- Spring 2020

Project 1: Tool for Quadratic Equations (I/O + functions + module + comments + if statements)

Due Date: **Deadline: see website, class policy and moodle for submission**
This is an individual project (discussions are encouraged but no sharing of code)

Description

The goal of this project is to practice basic Input-Output (I/O), basic functions, and conditional statements. This project contains two `.py` files:

- `project1.py`: it contains the main program that you will need to execute; this file is already completed (it is then **not allowed** to modify it).
- `functions_quad.py`: (module) it regroups all the functions called by the main program (this is a file that you must complete).

The project is designed to be incremental, you can then debug, test and run your code after each new task/option is implemented. Use your preferred IDE to read, write and save your files. If you are not using IDLE, however, make sure your program is running using the command prompt (your grade will be reduced by 50% if this is not the case or if the graders have to go the extra-step to make sure it runs using the command prompt). Do not forget to comment your code. Make sure to obtain the **exact same output** for the **exact same input** for the examples below (this includes syntax, blank spaces, and skipping blank lines). Your program will be tested with different inputs by the graders.

Submission/Grading Proposal

Since the file `project1.py` must stay unchanged, you will only need to submit your completed `functions_quad.py` file on Moodle. This project will be graded out of 100 points:

1. Your program should implement all basic functionality/Tasks and run correctly. (90 points)
2. Overall programming style: program should have proper identification, and comments. (10 points).

How to start

Although all the functions in `functions_quad.py` are incomplete, there are currently functional. It means that you should be able to run the main program `project1.py` without expecting a crash. By executing `project1.py` you will then get the following:

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 3
Enter value for c: 1

-Equation is: not done
-Derivative is: not done
-Extremum is: not done at x0=-0.75 and f(x0)=-111
-Discriminant is: -111
-Solving for f(x)=0
not done

-----------------Thanks and come back soon!----------------------
```

This program asks first the user to enter a value for `a`, for `b` and for `c` (we use 2, 3, 1, respectively, for this example). In the following, we are going to detail all the tasks that must be performed incrementally.

Task-1- [10pts]

At first, you want to return the correct expression of the quadratic function `f(x)`:

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 3
Enter value for c: 1

-Equation is: f(x)=2.0*x**2+3.0*x+1.0
-Derivative is: not done
-Extremum is: not done at x0=-0.75 and f(x0)=-111
-Discriminant is: -111
-Solving for f(x)=0
not done

-----------------Thanks and come back soon!----------------------
```

How to proceed?

1. Complete the function `my_quad_equation` that should accept a,b,c and should return the equation (as a String). The input coefficients a,b,c are considered float.
2. Make sure to obtain the same output than the example.

Task-2- [10pts]

Next, you must provide the expression of the derivative:

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 3
Enter value for c: 1

-Equation is: f(x)=2.0*x**2+3.0*x+1.0
-Derivative is: f'(x)=4.0*x+3.0
-Extremum is: not done at x0=-0.75 and f(x0)=-111
-Discriminant is: -111
-Solving for f(x)=0
not done

-----------------Thanks and come back soon!----------------------
```

How to proceed?

1. Complete the function `my_quad_equation_derivative` that should accept a,b and should return equation of the derivative (as a String). The input coefficients a,b are considered float.
2. Again, make sure to obtain the same output than the example.

Task-3- [15pts]

You can now proceed with computing the extremum:

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 3
Enter value for c: 1

-Equation is: f(x)=2.0*x**2+3.0*x+1.0
-Derivative is: f'(x)=4.0*x+3.0
-Extremum is: Minimum at x0=-0.75 and f(x0)=-0.125
-Discriminant is: -111
-Solving for f(x)=0
not done

-----------------Thanks and come back soon!----------------------
```

A quadratic equation is mainly a parabola which goes through an extremum that could be a minimum if a>0 or a maximum if a<0. To convince yourself, you can plot the function using your preferred software or just click here
`https://www.wolframalpha.com/input/?i=plot+2*x%5E2%2B3*x%2B1`
the main program is already computing the $x = x_0$ position of the extremum. An extremum of a function is obtained when $f'(x) = 0$ (derivative), and basic math should give you $x = -b/(2a)$ that we call `x0` here. Thereafter, the code reports to you if it is a minimum or maximum, and then returns the value of the function at the extremum $f(x_0)$ (as you could see in the plot of the function for our particular example).

How to proceed?

1. Complete the function `info_extremum` that should return a String containing either `Minimum` or `Maximum`.
2. Complete the function `evaluate_quad_equation` that should accept the coefficients a,b,c and a variable x as arguments and return the value of f(x) as float.

Task-4- [5pts]

Let us continue with the computation of the discriminant.

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 3
Enter value for c: 1

-Equation is: f(x)=2.0*x**2+3.0*x+1.0
-Derivative is: f'(x)=4.0*x+3.0
-Extremum is: Minimum at x0=-0.75 and f(x0)=-0.125
-Discriminant is: 1.0
-Solving for f(x)=0
not done

-----------------Thanks and come back soon!----------------------
```

How to proceed?

1. a new function called `compute_discriminant` that will take `a,b,c` as inputs and return the value of the discriminant Δ as float, which is the value of $\Delta = b^2 - 4ac$.

Task-5- [25pts]

The program ends with computing and printing the solutions of $f(x)$.

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 3
Enter value for c: 1

-Equation is: f(x)=2.0*x**2+3.0*x+1.0
-Derivative is: f'(x)=4.0*x+3.0
-Extremum is: Minimum at x0=-0.75 and f(x0)=-0.125
-Discriminant is: 1.0
-Solving for f(x)=0
Two solutions: -0.5 and -1.0
Factorize form: f(x)=2.0*(x+0.5)(x+1.0)

-----------------Thanks and come back soon!----------------------
```

If $\Delta > 0$ like it is the case here, the program will return two real solutions

$$x_1 = \frac{-b+\sqrt{\Delta}}{2a}, \quad \text{and} \quad x_2 = \frac{-b-\sqrt{\Delta}}{2a}$$

If $\Delta = 0$ the program should return only one real solution $x_1 = -b/(2a)$ (which happened to be the extremum already calculated previously). For example, if you choose the inputs `a=0.5, b=2, c=2`, the output should look like:

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 0.5
Enter value for b: 2
Enter value for c: 2

-Equation is: f(x)=0.5*x**2+2.0*x+2.0
-Derivative is: f'(x)=x+2.0
-Extremum is: Minimum at x0=-2.0 and f(x0)=0.0
-Discriminant is: 0.0
-Solving for f(x)=0
One solution: -2.0
Factorize form: f(x)=0.5*(x+2.0)**2

-----------------Thanks and come back soon!----------------------
```

Once the program tells you if two or one solutions are obtained, it displays just after the factorized form of $f(x)$. For example if $x_1$and x_2 are solutions, the factorized form is $f(x) = a*(x-x_1)(x-x_2)$, if only x_0 is solution then it is $f(x) = a*(x-x_0)^2$.

Finally, if $\Delta < 0$, there is no real solution, using the inputs `a=2, b=1, c=2` (for example), the program should then return:

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 2
Enter value for b: 1
Enter value for c: 2

-Equation is: f(x)=2.0*x**2+x+2.0
-Derivative is: f'(x)=4.0*x+1.0
-Extremum is: Minimum at x0=-0.25 and f(x0)=1.875
-Discriminant is: -15.0
-Solving for f(x)=0
No real solutions

-----------------Thanks and come back soon!----------------------
```

How to proceed?

1. You need to complete the function `print_info_solution_quad` which accepts a,b,c and the discriminant as inputs (float), and does not have any return values. Instead, all the printing will take place inside this function.

2. Inside the function, you will need to use some conditional statements to differentiate between the cases $\Delta = 0.0$, $\Delta > 0.0$ and $\Delta < 0.0$.

3. To compute the solutions you need to use the square root function $\sqrt{x}$ known as `sqrt(x)`. Since Python does not provide a built-in `sqrt` function, we will import it from the Math module. At the root (top) of your `function_quad.py` file add the following instruction
`import math`
you will then be able to call the square root function of x (for example) as follows: `math.sqrt(x)`

4. You are allowed to add any other functions inside the `function_quad.py` file, if you feel the need for it.

5. As you can see, the factorize form can manage the sign nicely in the expression. For example, we do get `f(x)=2.0*(x+0.5)(x+1.0)` or `f(x)=0.5*(x+2.0)**2`, and not `f(x)=2.0*(x--0.5)(x--1.0)` or `f(x)=0.5*(x--2.0)**2`. You may need a few `if` conditions to build the correct string expression. Hint: we note that $x = -|x|$ if x is negative, and $x = |x|$ if x is positive. The absolute value function is a Python built-in function that can be called using `abs(x)`.

Task-6- [10pts]

If you made it that far it means that your program is executing correctly. From now on, we aim at handling some special cases ("corner" cases), similarly of what was asked about obtaining the correct factorized form in previous task.

First, if you decide to enter as input `a=1, b=1, c=2.5` (for example), most likely your output for the equation would currently look like:

```
-Equation is: f(x)=1.0*x**2+1.0*x+2.5
```

However, when the coefficients a and b are equal to 1.0, then there is no need to include them in the expression. What you should get instead is:

```
-Equation is: f(x)=x**2+x+2.5
```

In addition, if `b=0`, we really do not want to see:

```
-Equation is: f(x)=x**2+0*x+2.5
-Derivative is: f'(x)=2.0*x+0
```

This is what one should expect:

```
-Equation is: f(x)=x**2+2.5
-Derivative is: f'(x)=2.0*x
```

You need to change the code (function `my_quad_equation`) appropriately to make this happen.

Task-7- [10pts]

If you decide to enter as input `a=-3.5, b=-2, c=-3` (for example), most likely your output for the equation would currently look like:

```
-Equation is: f(x)=-3.5*x**2+-2.0*x+-3.0
-Derivative is: f'(x)=-7.0*x+-2.0
```

The $(+-)$ is kind of ugly for both `f(x)` and `f'(x)`. We need to see the following instead:

```
-Equation is: f(x)=-3.5*x**2-2.0*x-3.0
-Derivative is: f'(x)=-7.0*x-2.0
```

In the cases where `a=-1 and b=-1`, we should also get (here `c=3.0`):

```
-Equation is: f(x)=-x**2-x+3.0
-Derivative is: f'(x)=-2.0*x-1.0
```

You need to change the code (functions `my_quad_equation` and `my_quad_equation_derivative`) appropriately to make this happen.

Task-8-[5pts]

If $\Delta < 0$ there is no real solutions but there are two solutions in the complex plane. The formula to compute these solutions is the same than the real formula above but now we need to evaluate the square root of a negative number. This can easily be done using the regular square root function since, if x is negative then $j\sqrt{|x|}$ is its square root (j stands for imaginary number). You can modify your program to display the complex solutions as follows for the case of `a=1, b=1, c=2`

```
--------------Welcome to the Quadratic Solver Tool----------------
                      f(x)=a*x**2+b*x+c

Enter value for a (a/=0): 1
Enter value for b: 1
Enter value for c: 2

-Equation is: f(x)=x**2+x+2.0
-Derivative is: f'(x)=2.0*x+1.0
-Extremum is: Minimum at x0=-0.5 and f(x0)=1.75
-Discriminant is: -7.0
-Solving for f(x)=0
Two complex solutions: (-0.5+1.32287565553j) and (-0.5-1.32287565553j)

-----------------Thanks and come back soon!----------------------
```

Task-9-[Bonus 5pts]

In your program you may use a conditional statement for comparing the value of Δ with 0.0. It is always difficult to compare float numbers. For example if you try `a=2.5, b=3.2, c=1.024`, it is easy to show that $\Delta = b^2 - 4ac$ should be equal to 0. Using Python, however, you will most likely obtain something like $\Delta = 1.7763568394002505e - 15$. The latter is also zero in floating arithmetic but this makes it difficult to use a conditional statement while comparing with the value 0.0. Find a way to solve this problem. Note that no TAs help/guidance is provided for bonus questions.