

# CSE102 – Computer Programming

## Homework #1

**Due Date: 27/02/2020**

Hand in: A student with number 20180000001 should hand in a file named 20180000001.c for this homework and compress it into a .zip file.

---

**Homework Description:** You will write a C file with the main function with three additional functions described below. Your Program will start calling part1, part 2 and part 3 functions in that order. For each part, you will receive the inputs from the user and print the output to the console. Details of the parts are further discussed below. Please pay attention to the output format. Any deviation from the shared format may be penalized regardless of the correct execution.

**Part 1.** [30pts] Your function will read three integers from the command prompt as coefficients of a quadratic equation and calculate the roots of the equation. If the roots are Real then Your program will print the equation and the roots to the console in descending order. If roots are complex numbers then the program will inform the user about the complex roots.

**Example 1:**

```
Please Enter the first coefficient:1
Please Enter the second coefficient:-3
Please Enter the third coefficient:-4
Output >Your equation 1x^2 - 3x - 4 have real roots {4,-1}.
```

**Example 2:**

```
Please Enter the first coefficient:1
Please Enter the second coefficient:0
Please Enter the third coefficient:1
Output> Your equation 1x^2 + 0x + 1 does not have any real roots. The function prototype is:
Function prototype is void find_root()
```

**Part 2.** [40pts] Your second function will search for the roots by a numerical method called "Newton's Algorithm". This method is an iterative root-finding algorithm that produces better approximations to the roots of a real-valued function. You can find a very easy-to-understand introduction at **Wikipedia**. Furthermore, web page at <https://planetcalc.com/7748/> can serve u as both a guideline and a validation resource.

In this function, you will again ask for three coefficients and an initial guess from the user and apply Newtons Method 5 times. In each step, you will compare the actual root and the estimated root (from the Newtons method) and print the difference. Unlike the first function, you don't have to worry about complex roots. **You can safely assume that users will not provide a function with complex roots.**

**Example:**

Please Enter the first coefficient:1  
 Please Enter the second coefficient:-3  
 Please Enter the third coefficient:-4  
 Please Enter the initial: 10

Output > Your equation is  $1x^2 - 3x - 4$  and iterations are

Step	x	f(x)	Difference
x1	6.1176	15.0726	2.1176
x2	4.4856	2.66366	0.4856
x3	4.0395	0.19899	0.0395
x4	4.0003	0.00154	0.0003
x5	4.0000	0.00000	0.0000

Function prototype is `void find_newtonian_root()`

**Part 3.** [30pts] Given two integer you will calculate, divisibility of the first integer by the second one. If it is not, you will find the closest divisible integer. You will ask for the user input in the main function and pass the parameters to the `find_multiple` method. This function will return to the main function and it will be printed on the command prompt.

**Example 1:**

Enter the first integer: 76  
 Enter the second integer: 15  
 Output> Closest number to 76 that is multiple of 15 is 75.

**Example 2:**

Enter the first integer: 76  
 Enter the second integer: 13  
 Output> Closest number to 76 that is multiple of 15 is 78.

**Example 3:**

Enter the first integer: 76  
 Enter the second integer: 19  
 Output> Closest number to 76 that is multiple of 15 is itself.

Function prototype is `int find_multiple_closest(int a, int b)`

**General Rules:**

1. Obey and do not break the function prototypes that are shown on each part, otherwise, you will get zero from the related part.
2. The program must be developed on Linux based OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.

3. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it's working in some way.
4. Upload your .zip file on to Moodle to deliver your homework. The zip file must consist of one .c file that contains your solutions. Name format can be found on the top of this homework sheet.
5. You can ask any question about the homework by sending an email to madede[@gtu.edu.tr](mailto:madede@gtu.edu.tr).