CSE102 – Computer Programming (Spring 2021)

Homework #1

Handed out: March 3, 2021.

Due: 11:55pm March 14, 2021.

Hand-in Policy: Via Moodle. No late submissions will be accepted.
Collaboration Policy: No collaboration is permitted.
Grading: This homework will be graded on the scale of 100.

Description: In this homework, you will write a complete C program implement several functions as described below. You are expected to reflect what you have learned in class up to this point.

You are provided with six separate files (in HW1_Src.rar):

- **main.c:** Contains the main function. You are not expected to modify this file in your submission. You may modify it for your own testing and debugging needs.
- **hw1_io.h:** Contains the declarations of input and output related functions for this homework. You are not expected to modify this file in your submission. You may modify it for your own testing and debugging needs.
- **hw1_io.c:** This file will contain your implementation of the functions declared in the associated header file. The details of these functions' behaviors are provided below.
- **hw1_lib.h:** Contains the declarations of integral and root finding functions for this homework. You are not expected to modify this file in your submission. You may modify it for your own testing and debugging needs.
- **hw1_lib.c:** This file will contain your implementation of the functions declared in the associated header file. The details of these functions' behaviors are provided below.
- **makefile:** This is a makefile provided for you to use for compiling and testing your code.

Assume that:

- **Degree 3 Polynomial**: is defined by four numbers as the coefficients of a degree-3 polynomial of the form:
$$p(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$
- **Degree 3 Polynomial**: is defined by five numbers as the coefficients of a degree-4 polynomial of the form:
$$p(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

The following provides the details of the functions to be implemented:

- **void write_polynomial3(double a0, double a1, double a2, double a3)**: Writes the polynomial in a pretty format. No sign replications should be allowed. For example:
```
write_polynomial3(-1.0,1.0,-3.2,0.0);
```
should print:
```
-x^3+x^2-3.2x
```

- **void write_polynomial4(double a0, double a1, double a2, double a3, double a4)**: Writes the polynomial in a pretty format like write_polynomial3.
- **double integral3(double a0, double a1, double a2, double a3, double xs, double xe, double delta)**: Numeric calculation of the integral of a degree three polynomial within the range [xs,xe]. The given parameter delta provides the step size for the integration.
- **double integral4(double a0, double a1, double a2, double a3, double a4, double xs, double xe, double delta)**: Numeric calculation of the integral of a degree four polynomial within the range [xs,xe]. The given parameter delta provides the step size for the integration.
- **double root3(double a0, double a1, double a2, double a3, double xs, double xe)**: Finds a root of the given 3rd degree polynomial within the range of [xs,xe]. You can assume that there is a root in the given range.
- **double root4(double a0, double a1, double a2, double a3, double a4, double xs, double xe)**: Finds a root of the given 3rd degree polynomial within the range of [xs,xe]. You can assume that there is a root in the given range.

Useful Hints: Here are some things that might make your development a bit easier.

- For testing your code use files for inputing data and getting the output. For example:

  ```
  $ hw1 < input.txt > output.txt
  ```

  will get the input from the file “input.txt” and will write the output to the file “output.txt”. This way you can easily make a lot of entries to test your code without using the keyboard again and again.
- Use the makefile to compile your code. You can add a run case to your makefile to do the compilation and testing with one simple make command.

What to hand in: You are expected to hand in a zip or rar file including the six files above. Your implementations should be completed in “hw1_io.c” and “hw1_lib.c”. The rest of the files should not be modified from their original versions.

- **HW1_lastname_firstname_studentno.rar / HW1_lastname_firstname_studentno.zip**

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 seydanurahi@gtu.edu.tr.