RBE104TC C/C++ Programming Language
Assignment 1

Contribution to the Overall Marks	40%
Issue Date	Sept. 11th 2023
Submission Deadline	Oct. 15th, 23:59, Beijing Time

Assignment Overview:

This assignment is geared towards assessing fundamental coding concepts in C/C++ and initiating the process of code development using the software development process (SDP) discussed in relevant lectures.

In composing the SDP report **[in English]**, we request you to fulfil the following criteria:

- Problem Formulation: Clearly outline the problem and assigned task.
- Analysis: Identify inputs, outputs, and any additional requirements.
- Design: Specify a sequence of steps (illustrated through flowcharts) necessary to accomplish the given task.
- Implementation: The C code must be submitted in a separate file. Please indicate the file name and provide instructions for user operation.
- Testing: Describe how you tested and validated your code for required functionalities, as well as its robustness.

You are expected to apply this approach to each of the provided simple exercises. Important notes to consider:

- Incorporate clear comments **[in English]** within your code to facilitate understanding.
- Detail your testing procedure and observations made during testing. Provide **screenshot images** along with comprehensive explanations to showcase actual implementation output.

Exercise 1 (50%):

Create a C/C++ program that accomplishes the following tasks:

- Read the following data from the keyboard and store it in suitable variables:
    - A complete name (e.g. John Smith)
    - A telephone number (e.g. 12345678900), assuming all phone numbers consist of 11 digits
    - A 2-digit decimal number (e.g. 22)
    - A temperature in degrees Celsius (e.g. 28.5°C)
- Divide the first 6 digits of the telephone number by the last 5 digits and save the result in a variable. Display this value on the screen (e.g. 123456/78900 = 1.56).
- Display an integer on the screen in decimal, octal, and hexadecimal formats (e.g. 45, 55, 2D).
- Convert the temperature from degrees Celsius to degrees Fahrenheit and degrees Kelvin. Store these values in appropriate variables rounded to the nearest whole number and display them on the screen (e.g. 10°C -> 50°F, 283°K).
- Determine the real roots of a quadratic equation for any real input values a, b, and c (e.g. handle cases of invalid input where a = 0, two equal real roots, two distinct real roots, or no real roots).

Exercise 2 (50%):

Write a function to search a character sequence pointed by a pointer (called “obj”), in another character sequence (called “source”). Return the pointer pointing to the found character. If there are more than one target found in source, return the pointer points to the first one.

Eg1: search for “C” in “ABCDEF”, return the pointer point to ‘C’.

Eg2: search for “Z” in “ABCDEF”, return a NULL pointer.

Eg3: search for “CD” in “ABCDEF”, return the pointer point to ‘C’.

Eg4: search for “CF” in “ABCDEF”, return a NULL pointer.

Eg5: search for “A” in “ABCAFC”, return the pointer point to the first ‘A’.

The function header is given by:

```
char *findC (char const *source, char const *obj);
```

What should be submitted?

You should submit the followings:

- A concise report (with text spanning a few pages) accompanied by C source codes. This report should delve into the specifics for each question:
  - Detail SDP steps 1 to 3 within the report (Problem Specification + Analysis + Design), accounting for 40% of the evaluation.
  - Encompass SDP step 4 (Implementation + Robustness) with your source code, incorporating comments. Implementation contributes 35% while Robustness adds 5%.
  - Elaborate on SDP step 5 (Testing), elucidating how you validated the correctness, robustness, and thoroughness of your codes. Employ screenshots and ample explanations as verification, encompassing 20% of the evaluation.
- For a comprehensive grading scheme, please consult the Marking Guidelines available on the Learning Mall system.
- The report must be saved in *.pdf format. As for C/C++ source code files, they should be compressed into a single archive file. Your final submission should comprise:
  - The report
  - Zipped source code files.
- The naming convention for the submitted Report and source code files should adhere to the subsequent format:
  - StudentID_LastName_FirstName_AssignmentID.pdf
  - StudentID_AssignmentNumber_ExerciseID.c/cpp
  - StudentID_LastName_FirstName_AssignmentID.zip

As an illustration:

The report and C source file would be denoted as follows:

- 1234567_Albert_Einstein_1.pdf
- 1234567_1_1.c
- 1234567_2_2.cpp

The ultimate zipped submission file would be named:

- 1234567_Albert_Einstein_1.zip

How the work should be submitted?

Submission should take place electronically through the Learning Mall system, enabling us to execute your software during the assessment. Additionally, feedback will be provided via the same Learning Mall system.