EEE101 C Programming and Software Engineering 1 – ASSESSMENT 2

Assessment Number	2
Contribution to Overall Marks	15%
Issue Date	16/10/2017
Submission Deadline	06/11/2017 at 0900 (9am)

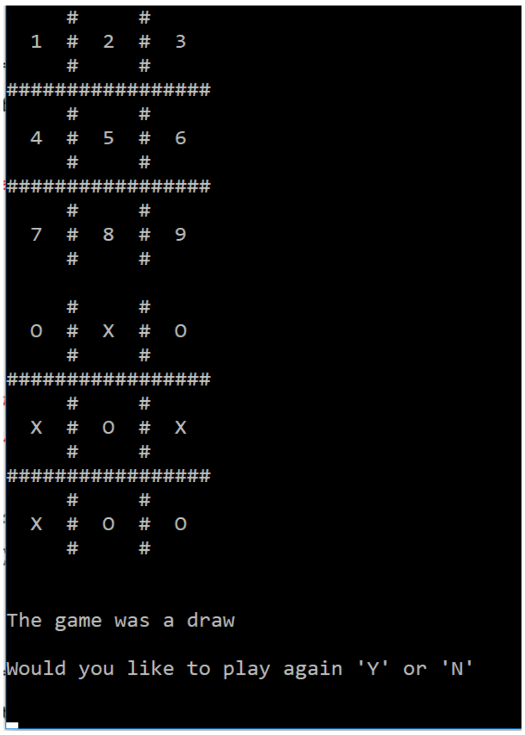
Assessment Overview

This assessment aims at testing some basic concepts of C programming, in particular, the use of functions to produce basic modular program structures. It also initiates the routine of code development using the software development process (**SDP**) presented in Lecture 1, namely the five main steps of the software development process:

1. Problem statement: formulate the problem.
2. Analysis: determine the inputs, outputs, variables, etc
3. Design: define the list of steps (the algorithm) needed to solve the problem.
4. Implementation: the C code has to be submitted as a separate file. Just indicate here the name of the file.
5. Testing: explain how you have tested and verified your C program.

Exercise

Write a modular C program that will allow two players to play the game of tic-tac-toe (also known as noughts and crosses), see the game example output in Figure 1.



```
     #     #
  1  #  2  #  3
     #     #
#################
     #     #
  4  #  5  #  6
     #     #
#################
     #     #
  7  #  8  #  9
     #     #

     #     #
  O  #  X  #  O
     #     #
#################
     #     #
  X  #  O  #  X
     #     #
#################
     #     #
  X  #  O  #  O
     #     #


The game was a draw

Would you like to play again 'Y' or 'N'
```

Figure 1: Example of game output

To start the game the two players should agree who will go first.

Players take turns to place their token (a ‘O’ or a ‘X’) on the board by choosing a number 1-9. The game is won if one player can place three of their tokens in a row, horizontally, vertically or diagonally. The game is drawn if all of the squares are filled and neither player has been able to make a row of three tokens.

Advice

This may initially appear very difficult, it is suggested that you first write down the steps that the program needs to perform and in what order (sequence), if necessary produce a flow chart and then think about how to do each part e.g:

- display the game board
- player 1 enters the number of the square they wish to choose
- the game board is updated
- the win status of the game is tested and the game finishes if the player has won
- player 2 enters the number of the square they wish to choose
- the game board is updated
- the win status of the game is tested and the game finishes if the player has won
- and so on…

Consider that for everything you want to repeat for example each iteration of the two players you will need a loop.

Produce the program one step at a time. e.g. if you can produce the input and test for player 1 then player 2 will be similar. Work out how to display your board (note it does not have to look exactly the same as in Figure 1). Each square choice can be a variable (i.e. 9 variables), that you can change the value of when it is chosen. Then the whole board can be printed with one printf statement that uses the values stored in the 9 variables.

The assignment tests your understanding of creating a basic modular program. Therefore, repetitive actions such as printing the board or testing for a win should be performed in separate functions and called when required. In this assignment use of global variables is acceptable.

Example Code

To help you get started, a program for a number guessing game is available on ICE, see character_game.c. Note that this is only to offer you ideas for writing your own program. You should not try to adapt the code to produce your work, this would be more difficult that writing your own.

What should be submitted?

You should submit the followings:

1) A short report (up to a few pages of text) detailing for each question:

   a) SDP steps 1 to 3 in the report (Report + Specification + Analysis + Algorithm Design) (40%)

   b) SDP step 4 (Implementation(35%) + Robustness(5%)): your C source code including the comments. (40%)

   c) SDP step 5 in the report (testing): you will explain your testing methodology including: what you wanted to test, how you have tested it and the outcome of your tests. (20%). Note: you do not need to include screenshots of results.

   Please refer to the file "EEE101 Marking Guidelines Assignment 2" on ICE for a detailed marking scheme.

2) The report in Microsoft Word or pdf format and C source code of your implementation for each question should be zipped into a single file, i.e. the zip file will contain 2 files, one document and one source code. (It is a good practice to include comments in your code stating the aim of the program, what are the inputs, what are the outputs, which algorithm is used, who is the author and so on.)

The naming of Report (.docx or .pdf), Source Code (.c) and Compressed file (.zip or .rar)

- StudentID_LastName_FirstName_AssignmentNumber-QuestionNumber.docx or .pdf
- StudentID_ AssignmentNumber-QuestionNumber.c
- StudentID_LastName_FirstName_AssignmentNumber.zip or .rar

For example

- 10115085_Zhang_Hanqing_2.docx
- 10115085_2.c

Zipped together into:

- 10115085_Zhang_Hanqing_2.zip

How the work should be submitted?

Should be submitted electronically through ICE so that the marker can run your programs during marking. Feedback and your grade will also be given through ICE.

Remember that you are responsible for ensuring that your C code will run in Visual Studio 2013 and that if it does not without documentary explanation you may get a 0 mark for your implementation.