

**SCHOOL OF DIGITAL MEDIA AND INFOCOMM TECHNOLOGY
DIPLOMA IN INFOCOMM SECURITY MANAGEMENT**

Year 2

ST2512 PROGRAMMING IN SECURITY

~ ASSIGNMENT 1 ~

Plagiarism Warning Statement

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. Plagiarism is a serious offence and disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

Overview

This is a group assignment. Each group is formed by TWO students.
For this assignment you will be implementing a C program:

A text based game, 'word game', and please take note of the following:

1. This is a group assignment; each group can be made up with not more than two students. Each group has to work independently from other groups. (See the Plagiarism Warning Statement)
2. Your application must be submitted via blackboard by **30th of June 2016 05:00PM**.
3. Your C files should be named **<filename>XXXX_YYYY.c** (XXXX and YYYY are the last 4 digits of your student numbers). In case you choose to work on your own without a partner, your C file should be named **<filename>XXXX.c** (XXXX is the last 4 digits of your own student number).

Important: You source file must be able to be compiled in the Ubuntu environment.

4. You only need to submit **C source files**.

Some sample and supplementary source codes will be provided to you at Blackboard.

5. At the beginning of your source codes, add in comments, to state clearly of the names, student IDs and class of the two students whom are forming the group.
6. At the beginning of each function, add in comments, to state clearly of the purpose of the function, the input parameter(s) (if any), and the return value (if any).
7. To ensure the higher degree of readability, you may need to add in relevant inline comments to explain your program (usage / logic / assumptions).
8. Late Submission Penalties

- Standard DMIT late submission penalty is applied.
 - 50% marks reduction per day. 0 mark after 1 working day.

9. References (See the Plagiarism Warning Statement)

In case you are applying some codes / techniques that you have learned from external reference, you need to state it clearly in your source codes. E.g. A sorting routine in C from a web page, a string manipulation algorithm from a text book, etc.

Detail Program Requirement

The word game:

Objective of the game:

This is a two player's game. You may design your game to be played by two human players. Alternately, you can design your game to be played by one human player against the computer program itself.

After the human players have entered their names. The game begins by showing a randomly picked word from the internal dictionary. This is the so called 'current' word.

The program will prompt the two players alternatively to enter new word according to the game rules to become the 'current' word. Depending on the word the players have entered, the players will earn some points. The program will repeat this process until one of the termination events* happens.

At each iteration, the system shall display the latest scores of the two players earned in this game session.

*The game will be terminated when one of the players makes an invalid input, or chooses to quit the game.

Acceptance Rules for a new word:

- The minimum length of the word is 6 letters.
- All letters in the word must be in lowercases.
- It must be listed in the internal dictionary (based on the file wordlist.txt).
- It has not been entered at the same game session.
- The first letter of the new word must match with the last letter of the 'current' word.
- It cannot be ended with -ing-.

Scoring Rules:

Each player will earn points based on each accepted word, the point value is the same as the game, scrabble:

```
(1 point)-a, e, i, o, u, l, n, s, t, r.
(2 points)-d, g.
(3 points)-b, c, m, p.
(4 points)-f, h, v, w, y.
(5 points)-k.
(8 points)- j, x.
(10 points)-q, z.
```

For example : The word 'trouble' will earn t(01)+r(01)+o(01)+u(01)+b(03)+l(01)+e(01)
= 9 points

Below are a few sample screens of the application in action.

1. Right at the beginning of the game.

```
st2512@ubuntu:~/myCprog/assignment/sample$ ./sample
+-----+
|Welcome to ST2512 playground!|
+-----+
Player1 Name Please=>wizkid
Player2 Name Please=>AlphaGone

round    1: wizkid scores :    0 AlphaGone scores :    0
current word is emptiest.

wizkid , please enter the next word (or 'h' for help , 'q' to quit)=>█
```

The program first displays a welcome message then it prompts for the names of the two players. The program also keeps track of the number of rounds in play and the running scores.

In this sample run, the first randomly picked word is 'emptiest'.

After the randomly picked word is displayed, the program prompts the first player, 'wizkid', for the next word.

2. When the user enter 'h' at the prompt.

```
This is a two players game.

The game begins by showing the players a randomly picked word
The two players will take turns to enter a new word to replace the current word according to the game rules.
Repeat the above process to earn scores from the game.
The game will be terminated when one of the players has chosen to quit or entered an unacceptable word

Acceptance Rules for a new word:

.The minimum length of the word is 6 letters
.It is listed in the internal dictionary.
.It has not been entered at the same play session.
.The new word must begin with the last letter of the current word
.It cannot be ended with -ing-.

Scoring Rules:

Each letter in the new word earns points based on the following point values:

(1 point)-a, e, i, o, u, l, n, s, t, r.
(2 points)-d, g.
(3 points)-b, c, m, p.
(4 points)-f, h, v, w, y.
(5 points)-k.
(8 points)-j, x.
(10 points)-q, z.

Press <Enter Key> to exit help
```

An 'online' help menu will be displayed. This screen explains the objective and the rules of the game.

3. Here is another snapshot from another game session :

```
round    3: wizkid scores : 24 AlphaGone scores : 24
current word is yesterday.

wizkid , please enter the next word (or 'h' for help , 'q' to quit)=>youngest
y(08)+ o(01)+ u(01)+ n(01)+ g(02)+ e(01)+ s(01)+ t(01)= 16

round    4: wizkid scores : 40 AlphaGone scores : 24
current word is youngest.

AlphaGone , please enter the next word (or 'h' for help , 'q' to quit)=>teaspoon
t(01)+ e(01)+ a(01)+ s(01)+ p(03)+ o(01)+ o(01)+ n(01)= 10

round    5: wizkid scores : 40 AlphaGone scores : 34
current word is teaspoon.

wizkid , please enter the next word (or 'h' for help , 'q' to quit)=>█
```

The above shows a few rounds of players' actions. At round 4, the challenge word is 'youngest' and 'AlphaGone' has entered 'teaspoon'. In this case, the program has accepted 'teaspoon' as a valid word and it lets 'AlphaGone' earned 10 more points. Take note that, the details of the score evaluation **is required** to be displayed.

4. Below is an invalid word scenario. In this case, the word has been entered, 'kickboxer', is not found in the internal dictionary. The game penalizes the player by

clearing his score to 0 and ends the game. (You can make up your own rules on the penalty for invalid input, you may even allow to game to continue, ie. allow user to retry one or two more times.)

```
current word is teaspoon.

AlphaGone , please enter the next word (or 'h' for help , 'q' to quit)=>necessary
n(01)+ e(01)+ c(03)+ e(01)+ s(01)+ s(01)+ a(01)+ r(01)+ y(08)= 18

round    3: wizkid scores : 10 AlphaGone scores : 18
current word is necessary.

wizkid , please enter the next word (or 'h' for help , 'q' to quit)=>yesterday
y(08)+ e(01)+ s(01)+ t(01)+ e(01)+ r(01)+ d(02)+ a(01)+ y(08)= 24

round    4: wizkid scores : 34 AlphaGone scores : 18
current word is yesterday.

AlphaGone , please enter the next word (or 'h' for help , 'q' to quit)=>yardstick
y(08)+ a(01)+ r(01)+ d(02)+ s(01)+ t(01)+ i(01)+ c(03)+ k(05)= 23

round    5: wizkid scores : 34 AlphaGone scores : 41
current word is yardstick.

wizkid , please enter the next word (or 'h' for help , 'q' to quit)=>kickboxer
Sorry wizkid, the word you have entered is not valid

Final score at round    5:
wizkid scores :  0 AlphaGone scores : 41

Congratulation to AlphaGone , you are the champion!
and Thank You to all. See you again Soon.
```

Take note that, the details of the final score and the winning result **are required** to be displayed.

5. Here is a snapshot of the player enters a 'q' and the game stops normally.

```
round    3: wizkid scores :  9 AlphaGone scores : 12
current word is departure.

wizkid , please enter the next word (or 'h' for help , 'q' to quit)=>q

Final score at round    3:
wizkid scores :  9 AlphaGone scores : 12

Congratulation to AlphaGone , you are the champion!
and Thank You to all. See you again Soon.
st2512@ubuntu:~/myCprog/assignment/sample$
```

Wordlist Data file

- You need to place the wordlist.txt file at the working directory of the game program. This wordlist.txt contains all the word data for the program to build the internal wordlist.

util.h header file and util.c helper function files

- To assist your group to work on this assignment, two C source files, util.h and util.c are provided for you.
- The util.h header file contains useful struct definition and function prototypes:
- The util.c contains useful helper functions to let you build a dictionary (an array of linked list to store words).

skeleton.c file

- The skeleton.c contains a runnable but not yet fully functional solution. It demonstrates the usages of the helper functions defined in the util.c. You can use this skeleton.c as the starting point to design and implement your own solution.

Marking Scheme

Here is the marking scheme of this assignment

Category	Description	Type (Group or Individual)	Weightage
Proper Submission	.Fulfilling the submission requirement stated in this document	Group	5 marks
Readability	.inline comments /documentation .choice of the variable names .proper indentations	Group	5 marks
Proper scheme to keep track of the words have been used in the current game session.	.Shall not have any assumption or limitation on how many words can be used in one game session.	Group	10 marks
Programming techniques	.proper choice of programming constructs .proper choice of algorithms .proper breakdown of the program with sub functions .proper breakdown the solution into multiple source files	Group	15 marks

Word Validation	.proper validation of user input according to the game rules: .no repeated word .word cannot be too short .first letter of new word must match with the last letter of the current word .only accepts lowercases input .the word must be listed in the wordlist.txt	Group	15 marks
Scores Computation	.Display the current scores .Correctness of the computing according to the point value chart.	Group	5 marks
Program completeness	.Fulfilment of the given specifications .Display format .Runtime errors free	Group	25 marks
Innovation / Creativity	.Features to enhance the game play (Make it more fun/ exciting/ playable)	Group	10 marks
Written Report	outline .Your contributions .the problems you have encountered .the knowledge you have gained from this assignment	<u>Individual</u>	10 marks

Total: 100 marks

"Non-runnable submission will not get more than 30 marks."

~ The End ~