Developing Technical Software
Assignment 1(30Marks)
Due date is 14th November 2018 23:59hrs

You may be asked to demonstrate/explain your work to the tutor, if you are absent/unavailable or fail to demonstrate properly, zero marks will be awarded.

Please note, this is an individual task and it will be checked for plagiarism. All the involved parties will be penalised if any plagiarism is found.

Please visit https://goo.gl/hQ87zg for more details.

Instructions

1. This assignment contains 4 questions. Q1 is for 10 marks, Q2 is for 7 marks, Q3 is for 6 marks, and Q4 is for 7 marks. The total assignment is for 30 marks and see the detailed rubric attached at the bottom of this page.
2. **Pseudo Code** is mandatory for all the programs
3. Submit only one document (Single word document)
4. Copy and paste your code into **word document**
5. Copy and paste the screen shot of the output window in the same word document
6. Use only .doc, .docx extensions – no other format will be accepted for marking
7. Marks will be given for proper **indentation and comments**

Qn 1. A gaming company consists of many kind of games and each customer orders online to buy a game. The gaming company will search for the specific game upon customers request in the order of first come first served basis. The gaming company has the following requirements.

1. There are many kinds of games available (at least ten). They may have more than one game for each genre
2. The customers list is organised in a Queue as a first come first served basis
3. The gaming company should be able to retrieve the data of the last sold games.

Your program should consist of a linked list with the following features

a.) Insertion at the head
b.) Insertion at the tail
c.) Deletion from the head
d.) Deletion form the tail
e.) Deleting specific element from the list

The next part should be implementing a Stack inherited from the above linked list with the following features

a.) Pop
b.) Push
c.) Top

The next part should be implementing a Queue inherited from the above linked list with the following features

a.) Enqueue
b.) Dequeue
c.) Top

The program flow is as follows:

The games should be saved in a linked list. The list of games ordered by customers should be saved in a queue. You should take the order from the beginning of the queue and search for it in the linked list then delete it and put this game name in a stack to be able to retrieve in the last sold game.

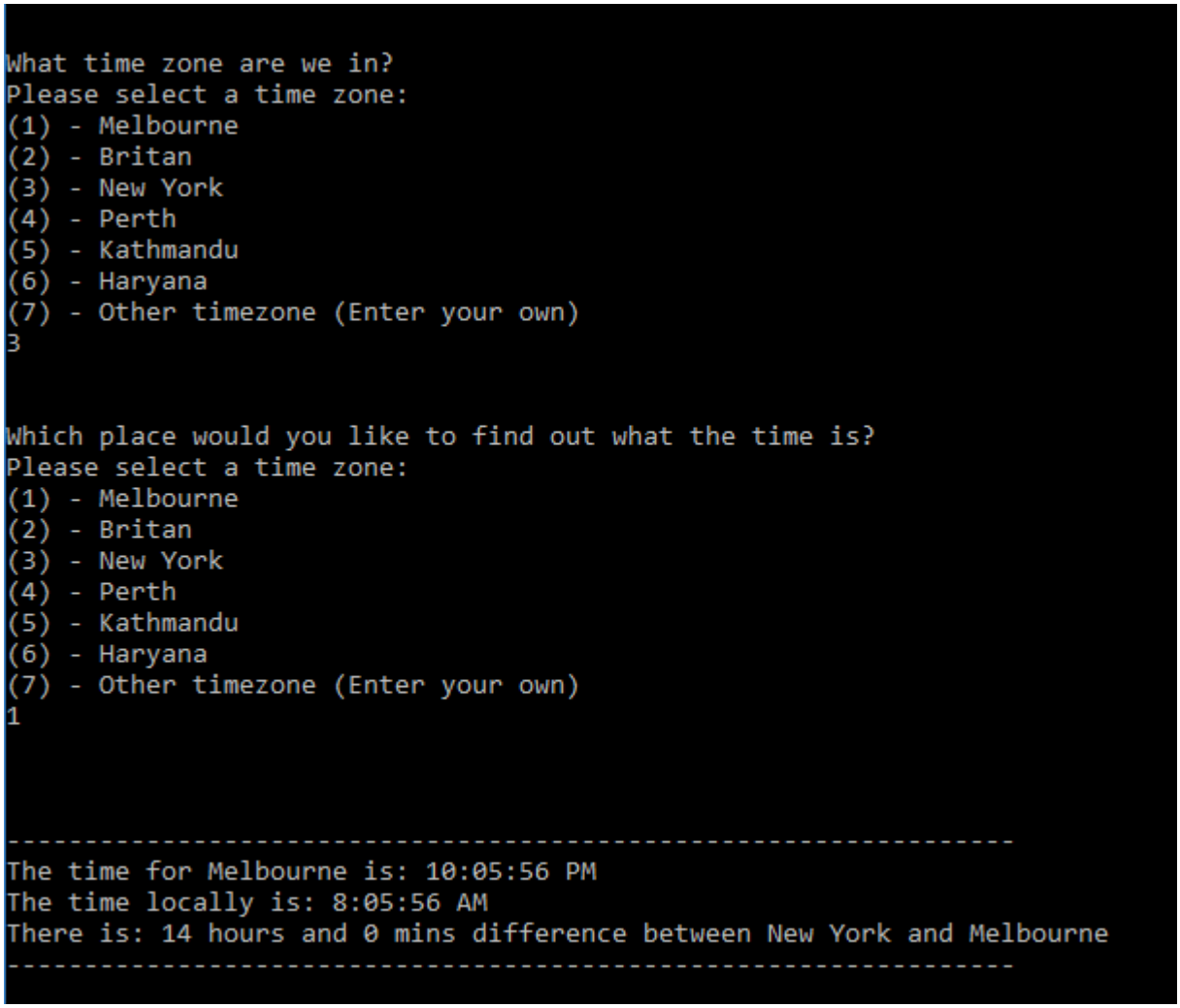
Sample screenshots of the output follows:

```
What would you like to do?
(1) - Display the current stock (linked-list)
(2) - Add a game to stock (linked-list)
(3) - Display next order info
(4) - Display all orders
(5) - Add order to que
(6) - Process the next order
(7) - Revers last order
(8) - Display info of last order
(9) - Quit Program
1



------------------------------
Game Name: Final Fantasy
Game Name: Forza Horizon
Game Name: Flight Simulator
Game Name: God of War
Game Name: Gears of War
Game Name: Guitar Hero
Game Name: Halo
Game Name: Metal Gear Solid
Game Name: Minecraft
Game Name: Sonic Mania
Game Name: Skyrim
Game Name: Tekken
Game Name: World of Warcraft
------------------------------

What would you like to do?
(1) - Display the current stock (linked-list)
(2) - Add a game to stock (linked-list)
(3) - Display next order info
(4) - Display all orders
(5) - Add order to que
(6) - Process the next order
(7) - Revers last order
(8) - Display info of last order
(9) - Quit Program
3



------------------------------
The next order is for:          Andrew
The game they ordered is:       Minecraft
------------------------------

What would you like to do?
(1) - Display the current stock (linked-list)
(2) - Add a game to stock (linked-list)
(3) - Display next order info
(4) - Display all orders
(5) - Add order to que
(6) - Process the next order
(7) - Revers last order
(8) - Display info of last order
(9) - Quit Program
```

```
(6) - Process the next order
(7) - Revers last order
(8) - Display info of last order
(9) - Quit Program
8


------------------------------
The last order was for: Steven
They ordered:           Sonic Mania
------------------------------

What would you like to do?
(1) - Display the current stock (linked-list)
(2) - Add a game to stock (linked-list)
(3) - Display next order info
(4) - Display all orders
(5) - Add order to que
(6) - Process the next order
(7) - Revers last order
(8) - Display info of last order
(9) - Quit Program
4


------------------------------
Order Nu.:1 is for:     Nathan
They ordered:           Tekken

Order Nu.:2 is for:     Manjil
They ordered:           World of Warcraft
------------------------------

What would you like to do?
(1) - Display the current stock (linked-list)
(2) - Add a game to stock (linked-list)
(3) - Display next order info
(4) - Display all orders
(5) - Add order to que
(6) - Process the next order
(7) - Revers last order
(8) - Display info of last order
(9) - Quit Program
1



------------------------------
Game Name: Final Fantasy
Game Name: Forza Horizon
Game Name: Flight Simulator
Game Name: God of War
Game Name: Gears of War
Game Name: Guitar Hero
Game Name: Halo
Game Name: Metal Gear Solid
Game Name: Skyrim
Game Name: Tekken
Game Name: World of Warcraft
------------------------------

What would you like to do?
(1) - Display the current stock (linked-list)
```

Qn 2. Write a C program that calculates the time difference between the system time and the users input time. For example if the user input is Brisbane time, the program should give the difference in hrs:mins:secs between the system time and Brisbane time. The program should use "structures" and both system time and user input time should be stored in structure variables.

Sample screenshot of the output follows:



Qn 3. Creating Linked List in C

Create a linked list using the following structure

```
struct studentid {

        int id;
        struct studentid *next;
};

typedef struct studentid STUDENTID;
```

*typedef STUDENTID * STUDENTIDPtr;*

You have to create a linked list manually similar to week 4 Qn2 lab exercise. There should be five nodes in the linked list and the elements should be the last 5 digits of your student id. One digit will go to one node and the node insertion should happen in order similar to week 4 Qn 1.

Eg: Assume your student id is 100989674, take the last five digits which is 89674 so the insertion order is

newptr=……..malloc(STUDENTID);
newptr->id=8;
.
.
.
9
.
.
6
.
7
.
.
4

and the final linked list should be in ascending order as shown below.

Qn4. Use the following structure for this task.

```
struct student{
        char name[30];
        int id;
        float mark;
};
```

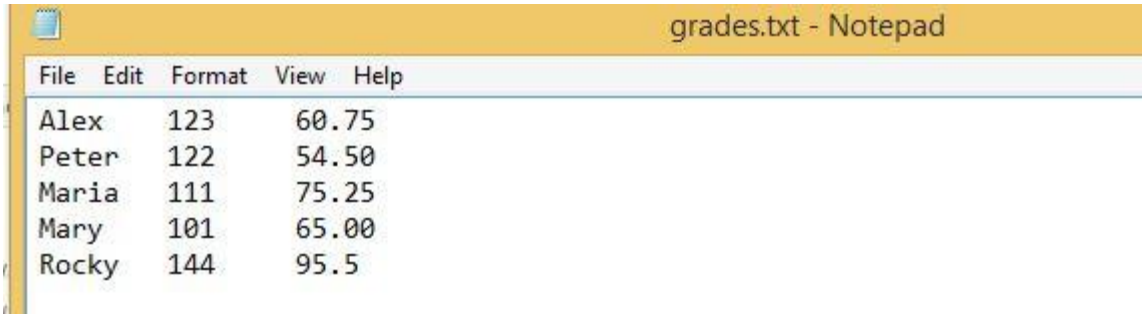
Declare an array named studentArray of the structure type shown above, the size of the array is limited to maximum 100.

The main() function handles all interactions with the **user** and other functions:

- It displays an appropriate welcoming message introducing the program.
- Calls a function named readFile() which opens the text file grades.txt for reading and stores all of the students details from the file to an array named studentArray. The grades.txt has three columns, first column contains name, second column contains id and third column contains mark. The readFile()function has two parameters: one for receiving the file variable and one for the array, both receiving arguments passed by reference.
- It then repeatedly calls the menu() function to display user options, get the user selection returned by the menu() function, use a switch statement to process user request by calling appropriate function(s)
- It displays the result with an appropriate message after processing user request.
- It displays a goodbye message when the user selects the **Quit** option from the menu and terminates the program.

The menu() function has no parameters. When called, it displays a menu of **4** options allowing the user to select one and returns this option to the calling main()function.
The options displayed should be:
(1) Display students' details
(2) Calculate average of all students' marks
(3) Add new student to the record
(4) Quit program

- **Option (1)** will use a function called displayStudents()called from the main()to display the contents of the studentArray array on the screen in an appropriate format. The displayStudents() function has two parameters: the array and the size of the array.
- **Option (2)** will use a function called calculateAverage() which is designed to calculate the average value of all marks in studentArray and return the result to the main() function which will then display it with an appropriate message. This function also has two parameters: the array and the size of the array.
- **Option (3)** will first use a function called updateFile() which will open the file in **append** mode, prompt the user for new student's name, id and mark, and then write the new data at the end of the file using the same format as the original file. It will then the call the readFile()function used in the beginning of the program again to read the contents of the updated file and repopulate the studentArray.
- **Option (4)** will terminate the program after displaying an appropriate goodbye message.
  Text file sample.



Good Luck!

COS10007-Assignment Rubrics

COS10007-Assignment Rubrics

Criteria	Ratings			Pts
Q1: Pseudo code	**1.0 Pts** Perfect and logical Pseudo code	**0.0 Pts** No attempt	**0.0 Pts** Partial Pseudo code	1.0 pts
Q1: A linked list of games with all the five required features	**2.0 Pts** All functions are working and screenshot of the output is included	**1.0 Pts** Some functions are working with respective output	**0.0 Pts** Partial attempt with no output	2.0 pts

COS10007-Assignment Rubrics				
Criteria	**Ratings**			**Pts**
Q1: The list of games ordered by customers are implemented in Queue with all the three functions	**2.5 Pts** **All functions are working and screenshot of the output is included**	**1.0 Pts** **Some functions are working with respective output**	**0.5 Pts** **Partial attempt with no output**	2.5 pts
Q1: Stack implementation(take the order from the top of the queue and search for it, then delete it and insert the game) with all the three functions	**2.5 Pts** **All functions are working and screenshot of the output is included**	**1.0 Pts** **Some functions are working with respective output**	**0.5 Pts** **Partial attempt with no output**	2.5 pts
Q1: Program flow	**1.0 Pts** **Perfect program flow as suggested in the question and screenshot of the output is included**	**0.0 Pts** **No program flow as suggested in the question**	**0.0 Pts** **Partial program flow as suggested in the question**	1.0 pts
Q1: Overall structure, indentation and appropriate comments	**1.0 Pts** **Perfect**	**0.5 Pts** **Almost Perfect**	**0.0 Pts** **Partial attempt**	1.0 pts
Q2: Pseudo code	**1.0 Pts** **Perfect and logical Pseudo code**	**0.0 Pts** **No attempt**	**0.0 Pts** **Partial Pseudo code**	1.0 pts

COS10007-Assignment Rubrics		
Criteria	**Ratings**	**Pts**
Q2: System time is stored in structure variables (Should not use system defined structures)	**1.5 Pts** Perfect and screenshot of the output is included / **0.5 Pts** Partial Attempt / **0.0 Pts** No attempt	1.5 pts
Q2: User time is stored in structure variables. User time captured from the keyboard	**2.0 Pts** Perfect and screenshot of the output is included / **1.0 Pts** Partial Attempt / **0.0 Pts** No attempt	2.0 pts
Q2: Correct calculation of time difference	**1.5 Pts** Perfect and screenshot of the output is included / **0.5 Pts** Partial Attempt / **0.0 Pts** No attempt	1.5 pts
Q2: Overall structure, indentation and appropriate comments	**1.0 Pts** Perfect / **0.5 Pts** Almost Perfect / **0.0 Pts** Partial attempt	1.0 pts
Q3: Pseudo code	**1.0 Pts** Perfect and logical Pseudo code / **0.0 Pts** No attempt / **0.0 Pts** Partial Pseudo code	1.0 pts
Q3: Used the same structure given in the question	**1.0 Pts** Perfect / **0.0 Pts** No attempt / **0.0 Pts** Partial Attempt	1.0 pts
Q3: Insertion is in the order of student id	**1.5 Pts** Correct order and screenshot of the output is included / **1.0 Pts** Partial Attempt / **0.0 Pts** No attempt	1.5 pts

COS10007-Assignment Rubrics		
Criteria	**Ratings**	**Pts**
Q3: Output is in the ascending order	**1.5 Pts** Yes and screenshot of the output is included / **1.0 Pts** Partial Attempt / **0.0 Pts** No attempt	1.5 pts
Q3: Overall structure, indentation and appropriate comments	**1.0 Pts** Perfect / **0.5 Pts** Almost Perfect / **0.0 Pts** Partial attempt	1.0 pts
Q4: Pseudo code	**1.0 Pts** Perfect and logical Pseudo code / **0.0 Pts** No attempt / **0.0 Pts** Partial Pseudo code	1.0 pts
Q4: Proper readFile() and display () functions	**2.0 Pts** Proper functions are used. Also array of structure is used for display() along with screenshot of the output / **1.0 Pts** Partially correct / **0.0 Pts** No attempt	2.0 pts
Q4: Proper CalculateAverage() function	**1.5 Pts** Proper function is used along with screenshot of the output / **1.0 Pts** Partially correct / **0.0 Pts** No attempt	1.5 pts
Q4: Proper update() function	**1.5 Pts** Proper function is used along with screenshot of the output / **1.0 Pts** Partially correct / **0.0 Pts** No attempt	1.5 pts

COS10007-Assignment Rubrics				
Criteria	**Ratings**			**Pts**
Q4: Overall structure, indentation and appropriate comments	**1.0 Pts** **Perfect**	**0.5 Pts** **Almost Perfect**	**0.0 Pts** **Partial attempt**	1.0 pts
Total points: 30.0				