

## Developing Technical Software Assignment 2 (25 Marks)

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 3 questions. Q1 is for 10 marks, Q2 is for 7 marks, and Q3 is for 8 marks. The total assignment is for 25 marks and refer to the detailed rubric given in Assignment page for mark allocation.
2. Submit one-word document and a zip file. Use the following format to prepare the word document (use the report template available in assignment page).
  - a. Question No. (No need to copy and paste question)
  - b. Problem description, data table, pseudocode or flow chart etc
  - c. C/C++ program - copy and paste your source code (not the screenshot of the code)
  - d. Enough screenshots of the output that shows all possible outcome.
3. Marks will be given for proper **indentation and comments**.
4. **Assignment Demonstration** is mandatory.

### Other requirements:

- This assignment must be written in C for questions 1 and 3 and in C++ for question 2.
- Your code must have appropriate header (multiline/block) comments including your name and student number, the name of the .c/C++ file, the purpose of the program, brief explanations of variables and explanations of any code, which is not obvious to another programmer, summarising the input, output and local variables as well as expressions used in your program and test data.
- Include inline (single line) comments throughout the program describing important statements.
- Use appropriate and descriptive variable following the naming rules and conventions.
- Write a brief (no more than several pages) report, which illustrates your program design (algorithm or flowchart, identification of variables, constants) and include evidence of testing – screen shots or pasted output text of several tests, and the contents of the .c/C++ file.
- Marks will be allocated depending on the amount of original work submitted. Marks will be deducted for plagiarised and/or un-attributed work.

### Qn 1. Write a complete C program (10 Marks)

Create a **text file** that contains **four** columns and **ten** rows. First column contains strings values, second and third column contains integer values, and fourth column contains double values (you are free to use your own values).

Declare a structure that contains 4 elements (you are free to use your own variables).

First element should be a **char array** – to read first column values from the text file.

Second element should be an **int** value – to read second column values from the text file.

Third element should be an **int** value – to read third column values from the text file.

Fourth element should be a **double** value – to read fourth column values from the text file.

Declare an array of this structure with **size 10** and read the contents of the text file into this array.

Then prompt the user with the following instructions:

1: Display the details of the array – call a function to display the contents of the array on screen.

2: To sort the array (you should call sort **function** – output of the sorting function should be written onto a text file and terminal)

You should give the user the chance to sort in ascending or descending order with respect to **string** value.

Then you should give the user the option to select from different sorting techniques (you should write minimum two sorting algorithm functions, call the functions according to the choice the user enters – call the sorting function only after the user selects the above-mentioned options).

3: To search for a **string** element (Write the output on terminal)

You should give the user to select the searching technique (linear or binary – must use recursive version of the searching functions)

if binary is selected call a *sort* function first.

4: To insert these array elements into a linked list in the **order of string values**. Display the contents on the terminal.

5: Quit

Your complete program should have multiple files (minimum two .c files and two .h files).

Give your file name as heading and then paste your code. You must include your text files as well in the submission.

### Qn2. Write a complete C++ program (7 Marks)

Some credit card companies use [Luhn algorithm](#) to detect invalid credit card numbers. The key thing in this algorithm is to find a check digit, which should be the rightmost digit in any credit card number.

In this task, you must implement a version of Luhn algorithm using the instructions given below.

#### Step1: Read in credit card number as a series of digits into an array

Ask the user to enter series of one-digit positive numbers into a one-dimensional array (you must use an integer array for this problem), the number -1 is used to indicate the end of series, no need to read in -1 into the array. The maximum size of the array is set to 20.

**Step2: Find sum1 (main should call a function passing array and size)**

Ignoring the last *check digit* (right most digit) of the credit card number, and moving left, double the value of every second digit and find the sum of these doubled numbers. If the result of the doubling operation is a two-digit number, you should add the digits of the doubled number before finding the sum. You must printout the numbers for sum1, these should be in the order (from left to right) as it appears on credit card.

**Step 3: find sum2 (main should call a function passing array and size)**

Find the sum of all other numbers (last digit is not included in this sum as well). You must printout the numbers for sum2, these should be in the order (from left to right) as it appears on credit card.

**Step 4: Calculate check sum**

Compute the total of sum1 and sum2 and multiply the result by 9, checksum is found by extracting the rightmost digit.



Image courtesy: <https://www.nab.com.au/business/business-cards/nab-rewards-business-signature-card>

4	3	3	6	8	7	1	2	3	4	5	6	7	8	9	0
4+4=8		3+3=6		8+8=16 1+6=7		1+1=2		3+3=6		5+5=10 1+0=1		7+7=14 1+4=5		9+9=18 1+8=9	sum1=8+6+7+2+6+1+5+9 = 44
	3		6		7		2		4		6		8		sum2 = 3+6+7+2+4+6+8 = 36
															Total=44+36 = 80

$80 \times 9 = 720$  ← 0 is the check sum digit.

**Qn3. Write a complete C program - you must declare and use a self-referential structure for this question (8 Marks)**

You are asked to write a c program that deals with the online purchase of TVs. The purchase order is handled first come first served basis. When a customer places an order to buy a TV, the program should search that item in the data base and if it is found the purchase should be processed according to the instructions given below.

TVs available in the shop are kept in a text file called *TVs.txt*.

At the beginning of the program the details of the TVs are read from *TVs.txt* and stored in a LinkedList. Then the list of the TVs ordered by customers is saved in a queue. Then the program should take the order from the beginning (head) of the queue and search for it in the LinkedList. If the item found in the list delete it and put this TV in a stack to be able to retrieve for the last TV sold.

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

- Calls a function named readTVs() which opens a text file TVs.txt (a sample text file is shown below) for reading and storing names of TVs from the file to a LinkedList in order of name (insertion should happen in alphabetical order).
- It then repeatedly calls the menu () function to display user options, get the user selection returned by the menu() function, use a switch (or if ..else if) statement to process user request by calling appropriate function(s).

Details of options in menu function:

- (1) Display the current stock of TVs– here you display the contents of the LinkedList
- (2) Add a new TV to stock – you need to insert a new TV to LinkedList
- (3) Display next order information – displays the next TV in the order-list (first node of the queue)
- (4) Display all orders – displays all nodes of the queue
- (5) Add order to queue – adds new order to the end of the queue
- (6) Process the next order – Processes the first order in the queue. This function searches for this TV in the LinkedList and deletes (if found) from LinkedList and puts it into a stack, deletes from the queue as well.
- (7) Cancel last order – It cancels the last processed order. It inserts the TV (top of the stack) back into LinkedList (TV is not added back into queue).
- (8) Display info of last order – displays the information of the last processed order (top of the stack).
- (9) Update TV file – updates the TVs.txt with the remaining TVs in the list (including the TVs added in option2).
- (10) Quit program

Sample *Tvs.txt* file

```

TVs.txt - Notepad
File Edit Format View Help
101 Hisense A7G Smart TV
102 LG Self Lit OLED
103 Samsung BU8000 Crystal UHD
104 Sony X90J Bravia XR
105 Google Chromecast with Google TV
106 Samsung Q80B
107 TCL C825 4K Mini LED
108 Samsung The Frame
109 Sony XR X90J
110 Philips 8215 Series

```

**Assignment submission information:**

Submissions through **Canvas** must be made on or before the due date/time.

Each submission should have two files.

1. A report (name of the report should be with your student number, eg: 1012546\_assignment\_2.docx) – use template provided with this assignment.

This report will be used for plagiarism check using turnitin software. **20% of marks will be deducted if this report is missing for plagiarism check.** Report must (.doc/docx) contain:

- Description of the problem, table etc.
- A copy of the contents of the .c file for all tasks (copy and paste the code not the screenshot of the code).
- Pasted **text output** or **screen shots** of the working program resulting from the testing of the program.

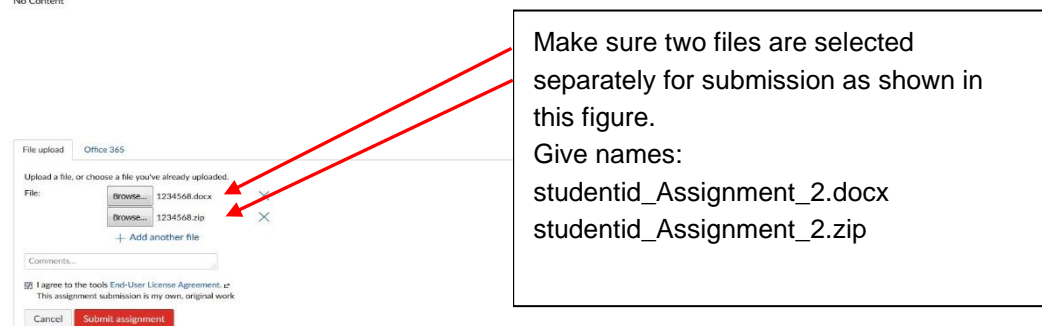
2. A .zip file (name of the zip file should be your student number, eg: 1012546\_assignment\_2.zip) containing:

- a) The actual programs (c/C++ source codes) with comments. Programs must be named studentid\_A\_Qn1.c, studentid\_A\_Qn2.c and so on.

## Assignment 1

Due: 9 Apr by 23:59 Points: 100 Submitting: a file upload Available: until 14 Apr at 23:59

No Content



Make sure two files are selected separately for submission as shown in this figure.

Give names:  
studentid\_Assignment\_2.docx  
studentid\_Assignment\_2.zip

**Rubric – Check Assignment 2 page****End of Assignment**