Fundamentals of Programming
Assignment 2
(30 Marks)

You must 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 assignment, 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 7 marks, Q2 is for 7 marks, Q3 is for 8 marks, and question 4 is for 8 marks. The total mark for assignment 2 is 30. Refer to the detailed rubric given on the Assignment page in Canvas for mark allocation.
2. Submit a Word document and a zip file. Use the following format to prepare the Word document (use the report template available on the assignment page).

   a. Question No. (No need to copy and paste questions) – Write a problem description
   b. C++ program - copy and paste your C++ program (not the screenshot of the code)
   c. **Enough** screenshots of the output that shows **all possible outcomes**.
3. Marks will be given for proper **indentation and comments.**
4. **Assignment Demonstration** is mandatory.

Other requirements:

- This assignment must be written in C++.
- Your code must have appropriate header (multiline/block) comments including your name and student number, the name of the .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 variables following the naming rules and conventions.
- 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. (7 Marks)

Assume you are working as a programmer in a communication field. Your task is to alter the original data before transmitting it because of security reasons. All their data is an **integer** that contains the number of digits between two and eight (inclusive), thus the possible integers are between 10 and 99999999. You must read in an integer and complete the following stages.

Stage 1:

Your task is to modify the data for transmission according to the following set of rules.

1. If the number is a two-digit number you swap the positions of the numbers (eg: if the number is 56, the altered number is 65).
2. If the number is a three-digit number, digits at positions 1 and 3 are swapped. (eg: number is 123, the altered number is 321)
3. If the number is four digits or above the following rules apply.

Replace the first digit (the first digit is the rightmost digit in the number) with the remainder after the sum of that digit plus 1 is divided by 10, the second digit with the remainder after the sum of that digit plus 2 is divided by 10, third digit with the remainder after the sum of that digit plus 3 is divided by 10, fourth digit by the remainder after the sum of that digit plus 4 is divided by 10 and so on.

The position of the number is counted from **right to left.**

5	9	1	5	4	2	3	9

Position 8	Position 7	··················		Position 3	Position 2	Position 1

Stage 2: Extra protection (done only on 8 digits numbers – if the user chooses this option)

Eight-digit data contains more valuable information, so you are asked to enable extra protection if the user chooses to.

The extra protection is done on the **modified number in stage one** by swapping the digit in position 1 with the digit in position 8, the digit in position 2 with the digit in position 7, the digit in position 3 with the digit in position 6, digit in position 4 with the digit in position 5.

If the modified number in stage 1 is as shown below:

2	3	4	5	6	7	8	9

The number after stage 2 will look like this:

9	8	7	6	5	4	3	2

Stage 3: (Decoding)

Once the encoding stages are over, write the codes to recover your original data from the encoded data. Your program should be able to show both the encoded and decoded numbers in a single run. You can watch a video here to get the logic behind this question.

YouTube Video - https://youtu.be/k_02wM4v6To

Qn 2. (7 Marks) – You must use an int array for this problem.

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 have to implement a version of the Luhn algorithm using the instructions given below.

Step 1: Read in credit card number as a series of digits into an array.

Ask the user to enter a series of one-digit positive numbers into a **one-dimensional array**, the number -1 is used to indicate the end of the series, no need to read in -1 into the array. The maximum size of the array is set to 20.

Step 2: Find sum1 (main should call a function passing array and size)

Ignoring the last `check digit` (rightmost 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 print out the numbers for sum1, these should be in the order (from left to right) as it appears on the credit card.

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

Find the sum of all other numbers (the last digit is not included in this sum as well). You must print out the numbers for sum1, these should be in the order (from left to right) as it appears on the credit card.

Step 4: Calculate checksum.

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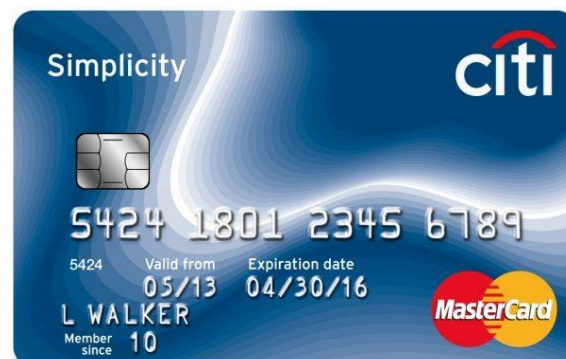

Image courtesy: http://echeck.org/citi-simplicity-credit-card-review/

5	4	2	4	1	8	0	1	2	3	4	5	6	7	8	9
5+5=10 1+0=1		2+2=4		1+1=2		0		2+2=4		4+4=8		6+6=12 1+2=3		8+8=16 1+6=7	sum1=1+4+2+0+4+8+3+7 = 29
	4		4		8		1		3		5		7		sum2 = 4+4+8+1+3+5+7 = 32
															Total=29+32 = 61

61 × 9 = 549

9 is the checksum digit.

Screenshots showing the working program:

Screenshot for the card number shown above.

Screenshot for the card number shown on the [wiki](#) page.

Qn 3. (8 Marks)

This question has two parts.

Part 1:

Write a complete program to draw rectangles on a SplashKit graphics window with width 1 and height equal to the values of the array elements. The first part of your program should have four functions, **main(), get_color(), fill_array(), and draw_bar()**.

```cpp
1   #include <iostream>
2   #include <cstdlib>
3   #include <ctime>
4   #include "splashkit.h"
5   #define SIZE 800
6   struct sample
7   {
8       int value;
9       color clr;
10  };
```

```
--------------------------------
Function: Main
--------------------------------
Local Variables:
- data (struct sample array with 800 (use SIZE) elements)

Steps:
1: Open a Window with title "Array rectangles" that is 800x700
2: Call srand() function
3: Call fill_array() passing data array
4: while (not quit_requested())
5:          Process Events
6:          Clear the Screen to COLOR_WHITE
7:          Call draw_bar() function passing data array
8:          Refresh the Screen limiting it to 60 FPS
```

```
--------------------------------
Function: get_color
--------------------------------
Parameters received – num (integer value)
Return type - color
Local Variables:
- clr (color)
- hue (float)
Steps:
1: set hue, the value of num devided with screen height (typecast for
   float value)
2: set clr, the value returned by hsb_color(hue,0.7,0.8)
3: return clr
```

```
--------------------------------
Function: fill_array
--------------------------------
Parameters received – data (struct sample array)
Return type - void
Local Variables:
- i (int for loop counter)

Fill the data array's value element with 800 random integers ranging
between 0 to 599 (you can take modulus with screen height - 100).

Fill the data array's clr element with color returned by getcolor
function (getcolor function is called by passing each array element)
```

```
--------------------------------
Function: draw_bar
--------------------------------
Parameters – data (struct sample array)
Return type - void
Local Variables:
- i (int for loop counter)
- x,y, rect_width (float values)
Steps
1: Set the rect_width the value 1
2: Start  a for loop that runs 800 times
3:          Set x, the value of rect_width times i
4:          Set y, the value Screen Height – data[i].value
5:          Fill Rectangle with data[i].clr, at position x,
            y, with width rect_width and height value part of data
            array (element of data array)
```

After implementing the above pseudocode in your C++ program, the output that you get will look as shown below.
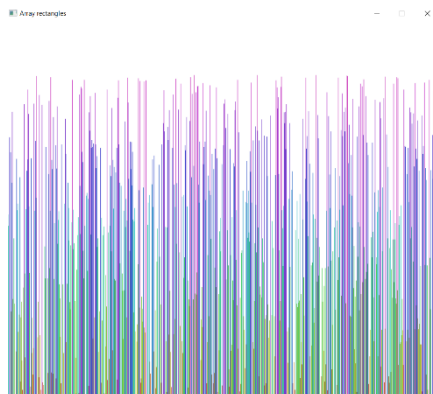
Part 2

Now extend your program to include three buttons on your window namely **Draw Bar**, **Sort 1**, and **Sort 2**. When you click on the Draw Bar button bars should be displayed on the window. When you click any of the sorts buttons the bars should be drawn after sorting the array and it should look as shown below.
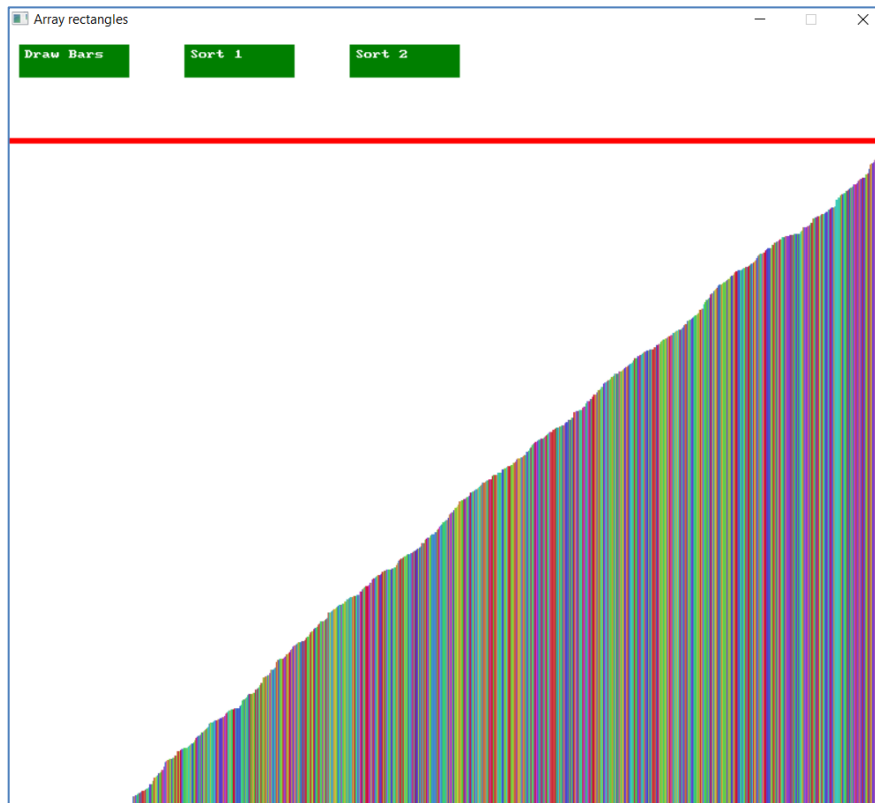
For this question, you need to implement two sorting techniques. One sorting technique can be selected from the week 7 lecture slides, and for the other, you are expected to conduct research and identify a suitable sorting algorithm. You can have any number of functions in part 2 as required. Watch this video to see the expected output for this problem.

Qn. 4 (8 Marks)

Write a complete C++ program to create a music player.

Your program should read several album names, each album has up to 5 tracks as well as a genre.

First, declare the genre for the album as an enumeration with at least three entries. Then declare an album structure that has five elements to hold the album name, genre, number of tracks, name of those tracks, and track location.

You can use the template given below, but highly recommended to use your own variable names and enumeration list.

```cpp
enum genre{ pop, Jazz, Classic};

struct album
{
    string album_name;
    genre kind;
    int track_number;
    string tracks[5];
    string tracklocation;
};
```

Static or dynamic memory can be used for the number of albums in this program.

In *main* you should have four options:

Option 1: call a function ***add_album*** – it allows the user to enter the album details.

Option 2: call a function named ***print_all_album*** to print out the album details.

Option 3: call a ***select_track_to_play*** function that allows the user to choose an album and then a track to play. It should print out:

"The track you selected " then the track name " from the Album: " then the album name " is now playing … from file location: " then the file location.

When the user selects a track to play your program must call an external program to play the track.

Option 4: Quit

```
Enter the option:
 1 to add an album
 2 to print the album details
 3 to play a track from an album
 4 to exit
1
Enter album name
Dangerous
Enter genre 0 -> pop, 1 -> Jazz, 2 -> Classic
0
Enter number of tracks in the album
3
Enter the names for these 3 tracks
BlackOrWhite
HealTheWorld
WhoIsIt
Enter the file location of these tracks
track_folder1

Enter the option:
 1 to add an album
 2 to print the album details
 3 to play a track from an album
 4 to exit
2

Name of the album  : Dangerous
Genre of the album : pop
No. of tracks      : 3
Tracks are :
BlackOrWhite
HealTheWorld
WhoIsIt
Tracks are located at track_folder1

Enter the option:
 1 to add an album
 2 to print the album details
 3 to play a track from an album
 4 to exit
1
Enter album name
Thriller
Enter genre 0 -> pop, 1 -> Jazz, 2 -> Classic
1
Enter number of tracks in the album
4
Enter the names for these 4 tracks
P.Y.T.
BeatIt
BillieJean
HumanNature
Enter the file location of these tracks
track_folder2
```

```
Enter the option:
 1 to add an album
 2 to print the album details
 3 to play a track from an album
 4 to exit
2

Name of the album  : Dangerous
Genre of the album : pop
No. of tracks      : 3
Tracks are :
BlackOrWhite
HealTheWorld
WhoIsIt
Tracks are located at track_folder1

Name of the album  : Thriller
Genre of the album : Jazz
No. of tracks      : 4
Tracks are :
P.Y.T.
BeatIt
BillieJean
HumanNature
Tracks are located at track_folder2

Enter the option:
 1 to add an album
 2 to print the album details
 3 to play a track from an album
 4 to exit
3
Select an album to play
Thriller
This album contains 4 tracks and those tracks are
1. P.Y.T.
2. BeatIt
3. BillieJean
4. HumanNature
Please select a track to play from the above list
BeatIt
The track you selected BeatIt.mp3 from the Album: Thriller is now playing from t
he location track_folder2

Enter the option:
 1 to add an album
 2 to print the album details
 3 to play a track from an album
 4 to exit
```

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_assignment1.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:
- o Description of the problem
- o A copy of the contents of the **.cpp** file for all tasks (copy and paste the code not the screenshot of the code).
- o Pasted **text output** or **screen shots** of the working program resulting from the testing of the program.
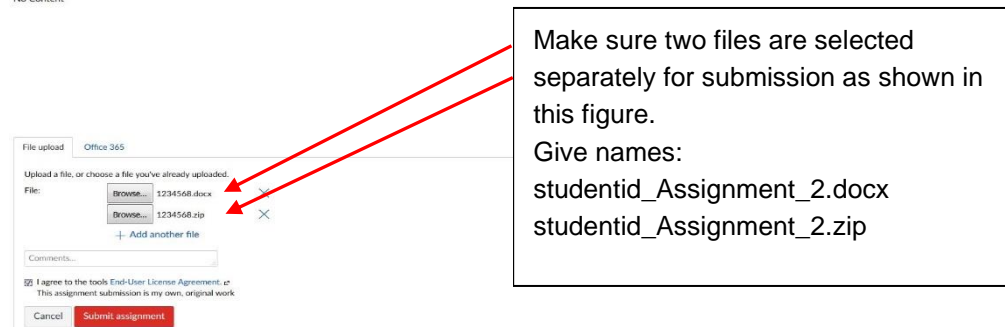
2. A **.zip** file (name of the zip file should be your student number, eg:
1012546_assignment_2.zip) containing:
- a) The actual programs (.**cpp** source codes) and all other relevant files. Programs must be named studentid_A_Qn1.cpp, studentid_A_Qn2.cpp and so on.

Make sure two files are selected separately for submission as shown in this figure.
Give names:
studentid_Assignment_2.docx
studentid_Assignment_2.zip

Marking Criteria – Refer to the rubric given on the Assignment page.

End of Assignment