EEET2482 Software Engineering Design
COSC2082 Advanced Programming Techniques
Semester 2, 2023

RMIT UNIVERSITY

LAB ASSESSMENT 1 (25%) – TEST QUESTIONS

Test Duration: 120 mins (+ 15 mins for submission)

NOTE: only submit **one .cpp file for each question** (two files for two questions), and **DON'T zip** them together.

Question 1 (10 pts)

Palindrome is a word, sequence or number that reads the same backward or forward, e.g., kayak, racecar, 11, 565. You are required to write a palindrome program using C++ with the requirements as follows:

(a) Write a function **sumDigit** that accepts an integer parameter, and returns the sum of all digits of the integer parameter. For example, if the integer parameter is 2234, then the sum of all digits of 2234 is 11 (i.e. 2+2+3+4 = 11).

(b) Write a function **checkPal** that checks an integer parameter and returns true if the sum of all digits of integer parameter is a palindrome and false otherwise. The function **checkPal** should call the function **sumDigit** when performing the summation of all digits of the integer parameter.

(c) Write a function **recPal** that accepts the sum from the **sumDigit** function, and returns a string that recommends addition or subtraction of values needed in order to have a valid palindrome (refer to Sample Run). This function is only called when the **checkPal** function returns false.

(d) Implement all functions in a C++ program that will take **an argument from the command line.**

(1) The program will read data from the file **data1.dat** (which is the first argument) and print out all information on the terminal. Save the output to a file named "**Result.txt**" as well. Make sure that the output is formatted properly in the file.

(2) Print out an error message, if the number of arguments from the command line is not correct.

Sample Run:

```
> ./a.exe data1.dat
2234: Result 11 is a palindrome.

451: Result 10 is not a palindrome.
Add 1 to make it a palindrome.

9: Result 9 is a palindrome.

10: Result 1 is a palindrome.

12345: Result 15 is not a palindrome.
Minus 4 to make it a palindrome.

9996: Result 33 is a palindrome.
```

Question 2 (15 pts)

Time class stores the hour and minute values of a time object into two private **int** based member variables: **hour** and **minute**. Write a C++ code to perform the following:

(a) Complete the public constructor in class **Time** that has two integer arguments of **hour** and **minute** as parameters to the object. When an argument is not supplied, or the argument supplied falls out of the allowable range, the constructor defaults to zero (0) for that particular data member. For example, if the **hour** supplied is 25, then it will be defaulted to zero (0) since 25 falls outside the allowable range of hour, which is from 0 to 23.

(b) Overload the addition (**+**) operator to this class to add two objects of **Time** class. While doing the addition, the function should handle overflow correctly (e.g. 55 minutes + 20 minutes = 1 hour 10 minutes) for each data member in the time object.

(c) Overload the output stream (**<<**) operator as a friend function to print out the **Time** object in formatted manner (**hour:minute**, e.g. 05:20). Use the functions from the **iomanip** library to format the output string as shown in the Sample Run.

(d) Create an array of 4 **Time** objects and read all information from a file named **data2.dat** (attached) to assign **hour** and **minute** values for them. Print out the time with the **lowest time** using the (**<<**) operator that you have written in Part (c). Work with the **hour** and **minute** values **directly** to find the **lowest time**, i.e. do not convert **hour** and **minute** values to a floating point value while determining the lowest time.

***Note**: Write a complete C++ class based on the aforementioned requirements. Your code must compile properly and function as per the requirements above.*

Sample Run:

```
05:23 + 23:42 = 05:05

Time 1 = 23:03
Time 2 = 10:30
Time 3 = 10:25
Time 4 = 18:12

Lowest time = 10:25
```

Part (d)

--- End of Paper ---