

LAB ASSESSMENT 1 (25%) – TEST QUESTIONS

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

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

Question 1 (6 pts)

Write a C++ program that prompts the user to input a string. The program should then identify and print the highest digit found within the string. If the string contains no digits, the program should display a message informing the user that no digits were found in the string.

Sample Run (bold denotes user's input) :

```
Enter a string: Hi572 45end  
The highest digit value is 7.
```

```
Enter a string: Hello there  
There's no digit in the string.
```

Question 2 (9 pts)

Write a C++ program which can work with command line arguments as follows:

Get two numbers in hexadecimal with an operator sign (+ or -), perform the operation indicated by the operator sign, and then print out the result in decimal. The hexadecimal number must be preceded by "0x" or "0X" (e.g. 0x12, 0xA). Your program should have the function **checkHex** that takes in one parameter **hexNum** of **string**, and returns true if **hexNum** is a hexadecimal number or false otherwise. Assume that the operator sign (+ or -) entered by the user is always correct.

Sample Run:

```
> ./q2 12  
Insufficient arguments.  
> ./q2 12 + 0xA  
Argument(s) provided is(are) not hexadecimal.  
> ./q2 12 - 0xAG  
Argument(s) provided is(are) not hexadecimal.  
> ./q2 0x12 + 0XA  
Result is 28  
> ./q2 0x64 - 0x1A  
Result is 74
```

Note: q2 is the executable file in this sample run.

Question 3 (10 pts)

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

- (a) Complete the public constructor in the **Task** class that has two input parameters: **t_Hour** and **t_Minute**, both of type **int**. When an argument (either **t_Hour** or **t_Minute**) is not supplied, the constructor defaults to zero (0) for that particular data member. Constructor also defaults to zero (0) if the **t_Minute** argument supplied falls out of allowable range of 0 to 59. For example, if the **t_Minute** supplied is 67, then it will be defaulted to zero (0).
- (b) Overload the operators so that we can do the following operations:
 - (i) greater than (>) operator: Compare two **Task** objects - **Task 1** and **Task 2**. Return true if **Task 1** > **Task 2**, else return false.
 - (ii) output stream (<<) operator: Display the **Task** object in the format "**hour:minute**" on the terminal, ensuring two-digit representation for both hours and minutes (e.g. "**05:09**" for 5 hours and 9 minutes, instead of "**5:9**". Refer to the Sample Run for clarity. **Hint:** Use the functions from **iomani** library to format the output.
- (c) Create an array of 4 **Task** objects and read all information from a file named **data.txt** (attached) to assign **t_Hour** and **t_Minute** values for them.
 - (i) Prompt the user to input the time taken for a task in hours and minutes. Then, display the user's task time and the times for the other 4 tasks from the file using the operator (<<) you implemented in Part (b)(ii). Assume that the user always provides a valid input within the allowable range of minutes.
 - (ii) Determine the number of tasks that have a time duration greater than the user's task time by using the operator (>) you implemented in Part (b)(i).

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 (bold denotes user's input):

```
05:09 > 33:42? False
```

```
Enter hours and minutes taken for your task: 10 40
```

```
Your task = 10:40
```

```
Task 1 = 33:03
```

```
Task 2 = 10:30
```

```
Task 3 = 46:25
```

```
Task 4 = 46:15
```

```
Number of tasks time greater than your task time: 3
```

--- End of Paper ---