Birla Institute of Technology and Science, Pilani,
First Semester 2022-23
CS F111 – Computer Programming
Lab Test - II

SET-B (Pink)

==========

05/02/2023 **Max. Marks: 75 M** **Duration: 120 mins**

==========

General Instructions

- This question paper comprises of one problem containing various sub-problems. The details of all these sub problems are described in the problem statement given in the next page.
- Read all the instructions and the problem statement very carefully before attempting the test.
- Carefully follow the submission instructions mentioned at the end of this document before uploading your solution on the **Dom Judge** portal.
- If you submit multiple submissions, only the latest one will be considered for evaluation. Whatever you submit on the Dom Judge portal will be considered as final. **It is your responsibility to make sure that you are submitting the right file**. **Also, ensure to save your file before you submit. And, it is your responsibility to make sure that your solution is properly submitted on to the Dom Judge portal.** Later on, if some student claims that he/she has submitted the solution and it doesn't appear on the Dom Judge portal, we won't be entertaining or listening to such issues.
- Programs (or functions) that have only the input/output (e.g., *printf*()/*scanf()* statements) without the key programing logic of the sub-problems, would **NOT** be considered for evaluation.

Instructions to attempt the test

Create a directory with the name **labtest2_<yourIDNumber> (Example: LabTest2_2022A1PS1234P)** in your home directory. Please download the files: **question paper pdf file** (which you are reading currently), "**WeatherDetails.h**", "**WeatherDetails.c**" and "**main.c**", from the domjudge portal. Copy these files into the directory which you have just created (**labtest2_<yourIDNumber>**). These are the three files you will be working upon. You should not make any changes to "**WeatherDetails.h**" and "**main.c**". You will have to complete the implementation of the functions given in "**WeatherDetails.c**". This is the only file that you should work upon to write your code answers. Also, you should not make any changes to the function parameters and the return types of the functions in "**WeatherDetails.c**". You should also not create any additional functions. If you do any of these don'ts you will definitely incur a penalty.

Carefully observe how each of the functions that you will have to implement, is invoked and used in the **main()** function (main.c), before attempting to implement them. Also, carefully observe the sample execution shown at the end of this question paper, while implementing each of the functions. Your functions should be implemented in such a way that the final compiled code when executed must give output similar to what is shown in the sample execution.

Problem Statement

You have to maintain a weather record for five days. The `weather details are` defined with the following attributes: {*Char array/String*: **city_name** (name of the city)**, weekdays** (which day of the week it is)}, and {*float:* **max_temp** (maximum temperature recorded)**, min_temp** (minimum temperature recorded)**, avg_temp** (average temperature)**, humidity**} in a structured manner. Please use the structure definition (**struct `weather`**) in **WeatherDetails.h** to store such records. The attribute **weekdays** can only store one of the following strings: **"*Sunday*"**, **"*Monday*"** and **"*Tuesday*"**. Complete the implementations of all functions in **WeatherDetails.c** with the following detailed specifications, given below:

(a) Implement function **`readWeatherInfo(int day)`** that reads the number of days as function parameter, and dynamically creates an array of size **day** that can store the weather records of **"day"** number of days. Then, using a loop you have to read weather records of **"day"** number of days from the user and store them in the above defined array. Then you should compute the value for **avg_temp** for each weather record using the **max_temp** and **min_temp** of that day (read from the user). Finally, this function should return the above created array. For simplicity, you can only read a single word (without spaces) for the attributes: **city_name** and **weekdays**. For example **"Pilani"** is a valid name to be stored, but not **"Pilani City"**. Follow the sample execution (given at the end of this document) carefully, before implementing the function. **15 M**

(b) Implement function **`printWeatherInfo(struct weather * wthr, int day)`** that receives an array of weather details (**`wthr`**), and number of days in that array (**`day`**), as parameters. Then, it prints the all the weather records in the array. Follow the sample execution (given at the end of this document) carefully, before implementing the function. **10 M**

(c) Implement the function **`createWeatherSLL(struct weather * wthr, int day)`** that receives an array of weather details (**`wthr`**), and number of days in that array (**`day`**), as parameters. This function dynamically creates (using **`malloc)`** a linked list using **`struct WeatherLinkedList`** for representing a linked list and **`struct WeatherNode`** to represent each linked list node. Please follow the structure definitions in **`WeatherDetails.h.`** Then this function *must then insert all the weather records contained in **`wthr`** array into the above linked list. The records must be stores in the reverse order*, i.e., the first record of **`wthr`** array should become the last node in the linked list, and the last record in **`wthr`** array should become the first node in the linked list. Basically, the order of the records should be reversed. Then, the function should return the linked list that was created. Follow the sample execution (given at the end of this document) carefully, before implementing the function. **15 M**

(d) Implement the function **`printWeatherSLL(struct WeatherLinkedList * newList)`** that receives a linked list (**newList**) and then prints the record stored at each node. Follow the sample execution (given at the end of this document) carefully, before implementing the function. **10 M**

(e) Implement the function **`printStats(struct WeatherLinkedList * newList)`** that receives a linked list (**newList**) and then prints those weather records stored in **newList**, that have their **max_temp** in the range **[26-40]**. The records must be printed in ***increasing order of their humidity values.*** **[Hint:** You have to create a separate array and storing those weather records of **newList** whose **max_temp** values are in the above defined range, by scanning the above linked list. Then sort the new array using selection sort and then print it**]**. Follow the sample execution (given at the end of this document) carefully, before implementing the function. **25 M**

You will have to additionally Create a script file "**myScript.sh**" in the **labtest_<yourIDNumber>** directory. Put all the commands that are required for compilation and linking to generate an executable with the name "**exe**".

Sample Execution of the Code

[Note: values in **pink** are user input for your clear visibility only.**]**

$ sh myScript.sh

Enter number of days for weather report: 5
Enter the Weather Details:
Enter City Name: Delhi
Enter Day: Monday
Enter Max Temperature: 25
Enter Min Temperature: 12
Enter Humidity: 85
Enter City Name: Jaipur
Enter Day: Sunday
Enter Max Temperature: 35
Enter Min Temperature: 29
Enter Humidity: 89
Enter City Name: Pilani
Enter Day: Tuesday
Enter Max Temperature: 25
Enter Min Temperature: 10
Enter Humidity: 75
Enter City Name: Jodhpur
Enter Day: Sunday
Enter Max Temperature: 32
Enter Min Temperature: 15
Enter Humidity: 90
Enter City Name: Goa
Enter Day: Monday
Enter Max Temperature: 38
Enter Min Temperature: 32
Enter Humidity: 92
###############

City Name: Delhi Weekdays: Monday Max Temp: 25.00
Min Temp: 12.00 Avg Temp: 18.50 Humidity: 85.00

City Name: Jaipur Weekdays: Sunday Max Temp: 35.00
Min Temp: 29.00 Avg Temp: 32.00 Humidity: 89.00

City Name: Pilani Weekdays: Tuesday Max Temp: 25.00
Min Temp: 10.00 Avg Temp: 17.50 Humidity: 75.00

City Name: Jodhpur Weekdays: Sunday Max Temp: 32.00
Min Temp: 15.00 Avg Temp: 23.50 Humidity: 90.00

City Name: Goa Weekdays: Monday Max Temp: 38.00
Min Temp: 32.00 Avg Temp: 35.00 Humidity: 92.00

###############

The Linked List is:

[City Name: Goa Weekdays: Monday Max temp: 38.00
Min Temp: 32.00 Avg Temp: 35.00 Humidity: 92.00] ==>

[City Name: Jodhpur Weekdays: Sunday Max temp: 32.00
Min Temp: 15.00 Avg Temp: 23.50 Humidity: 90.00] ==>

[City Name: Pilani Weekdays: Tuesday Max temp: 25.00
Min Temp: 10.00 Avg Temp: 17.50 Humidity: 75.00] ==>

[City Name: Jaipur Weekdays: Sunday Max temp: 35.00
Min Temp: 29.00 Avg Temp: 32.00 Humidity: 89.00] ==>

[City Name: Delhi Weekdays: Monday Max temp: 25.00
Min Temp: 12.00 Avg Temp: 18.50 Humidity: 85.00] ==>

NULL.

Printing the weather records whose max_temp in range [26-40] sorted by Humidity:

[City Name: Jaipur Weekdays: Sunday Max temp: 35.00
Min Temp: 29.00 Avg Temp: 32.00 Humidity: 89.00]

[City Name: Jodhpur Weekdays: Sunday Max temp: 32.00
Min Temp: 15.00 Avg Temp: 23.50 Humidity: 90.00]

[City Name: Goa Weekdays: Monday Max temp: 38.00
Min Temp: 32.00 Avg Temp: 35.00 Humidity: 92.00]

End of all execution

Submission Instructions

The directory **labtest_<yourIDNumber>** now contains "**WeatherDetails.h**", "**WeatherDetails.c**", "**main.c**" and "**myScript.sh**" files. Please upload **"WeatherDetails.c"** on the portal as your submission. You don't need to upload any other file. Make sure to save your file before uploading. You can also make multiple submissions of the same file.