

Birla Institute of Technology & Science, Pilani
Second Semester 2021-22 (New Academic Calendar)
CS F111 – Computer Programming
Online Programming Test

SET YELLOW

=====

24/07/2022

Max. Marks: 60M

Duration: 120 mins

=====

General Instructions

- This paper consists of only one question, which has various subparts whose details are described in the problem statement.
- Read all the instructions and the problem statement very carefully before attempting.
- Carefully follow the submission instructions mentioned at the end 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(s). Also, it is your responsibility to make sure that your solution is properly submitted on 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 any such request.

Instructions to attempt the test

Create a directory with the name **labtest_<yourIDNumber>** (Example: **labtest_2021A1PS1234P**) in your home directory. The zip file that you have downloaded (**Q1_SET_YELLOW.zip**) and unzipped contains the question paper (which you are reading currently), and a few C Program files: "**medicalstore.h**", "**medicalstore.c**" and "**main.c**". Copy these three files into the directory which you just created (**labtest_<yourIDNumber>**). These are the three files you will be working upon. You should not make any changes to "**medicalstore.h**" and "**main.c**". You will have to complete the implementation of the functions given in "**medicalstore.c**". This is the only file you should work upon and make changes to. Also, you should not make any changes to the function parameters and the return types of the functions in "**medicalstore.c**". 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 write their code. Also, carefully observe the sample execution shown at the end, 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.

The Problem Statement

You have to maintain a set of medicines. Each medicine has the following attributes: **ID** (Integer), **Name** (Char array), **Price** (float), **Quantity** (Integer). You can follow the structure definition (**struct medicine**) in **medicalstore.h**. Complete the implementations of the functions in **medicalstore.c** whose details are given below:

- (a) Implement the function **readMedicinesList()** that receives a **count** of medicines as a function parameter, and dynamically creates an array of size "**count**" that can store "**count**" number of medicines. Then, in a loop you will have to read the details (**name**, **price** and **quantity**) of those "**count**" number of medicines from the user. You should not be reading a value for **ID** of each medicine from the user. Rather, your program should assign a unique value to the **ID** starting from **1** for the first medicine, **2** for the second medicine, and so on. Finally, this function should return the array that you have just created. Assume that the name of every medicine is only a single word (Eg: "Xyza" and not "Biplex Forte"). Also while giving the input, restrict the price of a medicine to have upto two decimal places only (Eg: 23.75 is valid price but 52.6987 is not a valid price). **10M**
- (b) Implement the function **printMedicinesList()** that receives an array of medicines (**mItems**), and the number of medicines in that array (**count**), as parameters. It then prints the details of all the medicines present in that array. Follow the output shown in the sample execution (next page) carefully while implementing this function. **5M**
- (c) Implement the function **findMedicine()** that receives a float value (**medPrice**), an array of medicines (**mItems**), and the number of medicines in that array (**count**), as parameters. It then searches for the first medicine in the above array whose quantity is equal to **medPrice** and returns that item. If such an item is not found, it simply returns an empty medicine with its **ID = -1** and other fields uninitialized. Please follow how the function is called in the main function as well as the sample execution before implementing this. **10M**
- (d) Implement the function **findMinQuantityMedicine()** that receives an array of medicines (**mItems**), and the number of medicines in that array (**count**), as parameters. It then searches for the medicine in the above array that has minimum quantity and returns that medicine. If there are multiple medicines with that maximum price, any one of them can be returned. Please follow how the function is called in the main function as well as the sample execution before implementing this. **10M**
- (e) Implement the function **createMedLL()** that receives an array of medicines (**mItems**), and the number of medicines in that array (**count**), as parameters. It then dynamically creates a linked list using **struct mLinkedList** and **struct mNode** for defining/storing a linked list and linked list nodes respectively. Please follow the structure definitions in **medicalstore.h**. The linked list must store all the medicines of the **mItems** array in reverse order, i.e., the first medicine of the **mItems** array should become the last medicine stored in the linked list, the last medicine in the **mItems** array should become the first medicine stored in the linked list, etc. Basically the order of all the medicines should be reversed. Then this function should return the linked list just created. **15M**
- (f) Implement the function **printMedLinkedList()** that receives a linked list (**newList**) and then prints all the items present in it, while traversing the linked list from the first node to the last node. Follow the sample execution carefully before implementing this. **5M**

- (g) 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"**. Make sure that the name of the executable is **"exe"** only. **5M**

Sample Execution of the Code

[Note: Things in **bold red ink** are user entered values.]

```
jagat@XYZ:~/labTest$ sh myScript.sh
Enter number of unique medicines in the store:3

Enter details for Medicine 1:
Name:Xyzal
Price:20.54
Quantity:10

Enter details for Medicine 2:
Name:Zyrtec
Price:83.94
Quantity:25

Enter details for Medicine 3:
Name:Nexium
Price:75.96
Quantity:18

Medicine ID: 1, Name: Xyzal, Price: 20.540001, Quantity: 10
Medicine ID: 2, Name: Zyrtec, Price: 83.940002, Quantity: 25
Medicine ID: 3, Name: Nexium, Price: 75.959999, Quantity: 18

Enter the Price of the Medicine you wish to find: 83.94

The first medicine found with price 83.940002 is:
Medicine ID: 2, Name: Zyrtec, Price: 83.940002, Quantity: 25

The medicine with minimum quantity is:
ID: 1, Name: Xyzal, Price: 20.540001, Quantity: 10

Printing the Linked List created:
Medicine ID: 3, Name: Nexium, Price: 75.959999, Quantity: 18
Medicine ID: 2, Name: Zyrtec, Price: 83.940002, Quantity: 25
Medicine ID: 1, Name: Xyzal, Price: 20.540001, Quantity: 10
```

Submission Instructions

The directory **labtest_<yourIDNumber>** now contains “**main.c**”, “**medicalstore.c**”, “**medicalstore.h**”, and “**myScript.sh**” files. Create a zip file for this directory. The zip file should have the name **labtest_<yourIDNumber>.zip** (Example: **labtest_2021A1PS1234P.zip**). Upload this zip file as the solution to the Problem on the Dom Judge Portal.