**INSTRUCTIONS -** **PRACTICAL EXAM – CSD203**

**PLEASE READ BEFORE STARTING YOUR EXAM**

**=====================================================================================**

**Software Requirements**

* PyCharm, Notepad, Notepad++, Command Prompt, WinRAR / WinZip (or compress to ZIP file of Windows), Windows Explorer (File Explorer) on Windows 7 and above.

**Students are ONLY Allowed to use:**

* The softwares above and draft paper with pen/pencil.

**Instructions**

* Step 1: Students download the given materials from PEA Client.
* Step 2: Students read questions and prepare answers in the given template.
* Step 3: Submit a solution for each question:
  + The result is one folder for each question (Q1,Q2), which contains **.py** source files.

Example:

**+ Folder “Q1” only contains Q1.py for question 1**

**+ Folder “Q2” only contains Q2.py for question 2**

* + For each question, you must submit one folder as above to PEA Client.
* **Importance:** 
  + *Solutions will be marked by Automated Marking software.*
  + *The use of tools other than those allowed in the above section whether intentionally or unintentionally, is considered a violation of the exam rules, and the mark is 0*
  + *Do not: change the names of the folders, files, and struct (format) of .py files specified in the exam. If you change it, the marking software can not find the execute file (.py) or the program output to mark, thus the score will be 0*
  + *Do not: edit given statements in the main function. If you change, the marking software can not mark and the score will be 0.*
* ***Notes 01:*** 
  + *Do not edit given statements in the* ***main*** *function*
  + *You can create new functions if you think they are necessary.*
  + *Carefully read the instructions in each question to complete the practical exam.*
* ***Notes 02:*** 
  + ***The input and expected result in the questions are only used to test your codes.***
  + ***The input and expected result in the real testcases (for marking) are different from the given input and expected result in the examples below.***
  + ***Do not hardcode with the given expected result.***

**=====================================================================================**

**Question 1: (5 marks)**

**The given file Q1.py already contains statements to implement a simple program to monitor Drink objects using singly linked list structure. You should write statements to the following functions**:

1. **f1()**: **Insert at the beginning** of the current list a new Drink (named S1, given in the source code) such that: there is **no duplicated Drink’s code** in the list.

**Input**: size = 3

**Expected result**:

**2C014, Coca-Cola, 10, 600ml, 12.000**

PS021, Pepsi, 10, 390ml, 10.000

MD033, Mirinda, 45, 320ml, 12.000

SP005, Schweppes, 8, 320ml, 10.000

1. **f2()**: Write your code to **insert** a new node (given in the source code) **before the last node** of the current list.

**Input**: size = 5

**Expected result**:

PS021, Pepsi, 10, 390ml, 10.000

MD033, Mirinda, 45, 320ml, 12.000

SP005, Schweppes, 8, 320ml, 10.000

2C017, Coca-Cola, 20, 600ml, 15.000

**NEWNODE, Sprite, 15, 390ml, 12.000**

MD029, Mirinda, 14, 390ml, 18.000

1. **f3()**: Assuming that there is no duplicated amount in the list, find the node with **maximum amount**.

**Input**: size = 5

**Expected result**:

MD033, Mirinda, 45, 320ml, 12.000

1. **f4()**: **Remove the last node**, then **sort** the linked list in an **ascending** order according to Drink's **price**. (if two nodes have the same price, do not swap them)

**Input**: size = 7

**Expected result**:

PS021, Pepsi, 10, 390ml, 10.000

SP005, Schweppes, 8, 320ml, 10.000

MD033, Mirinda, 45, 320ml, 12.000

SP002, Bohuc, 18, 320ml, 12.000

2C017, Coca-Cola, 20, 600ml, 15.000

MD029, Mirinda, 14, 390ml, 18.000

1. **f5()**: Let’s define the ‘value’ of a node as follows: **value = amount\*price**. LValue and RValue of a node is the total value of nodes before and after that node (for example: if the list contains 7 nodes, and node K is the third node, so the LValue of K is the total value of two first nodes, and the RValue of K is the total value of the four last nodes).

Assuming the list contains at least 3 nodes, find the node satisfying that: **the difference of LValue and RValue is the minimum** (it also means that: |LValue - RValue| is the minimum).

**Input**: size = 4

**Expected result**:

MD033, Mirinda, 45, 320ml, 12.000