CS 509 - PG SOFTWARE LAB
Lab Assignment 1 Weightage: 20%
Submission Deadline: 22-August-2022 23:55pm

General Instructions:

- You must use only C, C++, Python or JAVA for this assignment.
- Prescribed specifications must be strictly followed. Failure to do so may lead to substantial loss of points.
- Make sure your code is well written (self explanatory variable names) and documented. You are likely to lose points if your TA cannot understand your code.
- You are not allowed to use any libraries. You are allowed to use only the primitive data types in the language such as arrays, structures, classes, hash_map, associative array, etc.

Question 1 (100 points): For this programming assignment, you are required to implement external sorting with the N-way merge algorithm covered in the class.

Dataset creation:

For this question, you would have to create a synthetic table (simulating sales records of department stores) containing 50000 records. Each record in this file contains four fields: (1) Transaction ID (an integer), (2) Transaction sale amount (an integer), (3) Customer name (string) and, (4) category of item. Transaction ID is an integer to identify each transaction uniquely in the dataset. You can create this field using a simple "counter" in your code. Transaction sale amount is a random integer between 1 and 60000. Customer name is a random 3 letter string. You can model this as a character array of length 3. Category of the item is a random integer between 1 --1500.

After creating this dataset, you need to simulate its storage on a disk. Define a disk block as a file which can store only **B** records of the synthetic table. Assume an unspanned organization i.e., records are not allowed to span across two disk blocks. Following this store your entire synthetic table as a collection of these "disk blocks." Each disk block (simulated as a file) should have a unique name, for that you can name them as 1.txt, 2.txt, 3.txt, …, etc. Basically, your original synthetic sales table would be stored as a series of files. For e.g., if B=300, then the first disk block (file) would store Row 1 -- Row 300; the second disk block (second file) would store Row 301 – Row 600. Do not hard code the value of **B.** It may be changed during the evaluation.

For sake of simplicity use text files for simulating the disk blocks. Also note that each disk block should have an entry at its end which stores the file name of the next disk block for the file (or the run). **You need to sort the file on the transaction sale amount. Note that all the intermediate runs (created during the sort process) should also be managed/stored as "simulated disk blocks" only. Finally, your code should produce a txt file as output which contains the records in the sorted order.**

Details of Simulated Main Memory:

Assume that you have space for storing only **M** "simulated disk blocks" in the main memory. In other words, you are allowed to read/operate on a maximum of **M** "simulated disk blocks" at a time. Do not hard code the value of **M**. It may be changed during the evaluation.

Evaluation details:

We will have following major test cases:

(a) Can the code dynamically change the order of merge (merging 2, 3, 4..M-1 runs) depending on the value of M given? Partial points will be given if the order of merge is hard coded to 2.

(b) Small dataset test case: Here, we will give you a small dataset (e.g., 20 records, B=1 M=3) (or ask you to create this small dataset) to test the correctness of your code.

(c) Large dataset test case: For this, we will test the code on the 50,000 records dataset to see the scalability aspects of your code. We may also change B and M values accordingly. Make sure your code is robust to these kinds of changes.