Birla Institute of Technology and Science, Pilani,
Second Semester 2023-24
CS F211 – Data Structures and Algorithms
Lab Test – I (Open Book)

==========

24/03/2024 **Max. Marks: 60 M** **Duration: 5hrs**

==========

General Instructions

- This question paper comprises one problem containing various sub-problems, whose details are described on the next page.
- Read all the instructions and the problem statements 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 make multiple submissions, only the latest one will be considered for evaluation.
- **It is your responsibility to make sure that you are submitting the right file**. **Also, ensure to save your file before you submit it.**
- **It is your responsibility to make sure that your solution is properly submitted to the Dom Judge portal. After submitting your solution, please download it and verify if it has been correctly uploaded or not. Repeat the submission if not correctly uploaded.**
- **Please get your submission verified by one of the invigilators before leaving the room.**

Instructions to attempt the test

Create a directory: **DSA_Labtest1_<yourIDNumber> (Example: DSA_LabTest1_2021A7PS1234P)** in your home directory. Create a file: "**LT1_Program_<yourIDNumber>.c**" in the above directory **(Example: LT1_Program_2021A7PS1234P.c)**. You will have to write your entire program within this file only. You will have to create structures and implement the functions as specified in the problem statements. You are free to choose their signatures, however, they should adhere to the specifications given in the problem statements. You are also allowed to declare any number of variables of any kind.

Carefully observe the sample execution shown at the end of this question paper. 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.

Marking Scheme

- 50% of the marks for execution and 50% of the marks for presentability (5%) and logic (45%) approximately.
- Well-documented and readable programs will get a better score for presentability.
- No marks for writing logically correct statements that are out of context.

Problem Statement

You need to construct a Binary Search Tree where each node stores a linked list. You can consider the first node of the linked list as the key to insert into an appropriate position in the BST. Given a file "file1.txt", where the first line contains how many data records are present in this file (say n). Then there are n lines in the file, where each line contains a space separated set of integers to be inserted as a linked list into the BST using the first integer value as the key. Once such a BST is built, you will first need to sort all the linked lists at each of the nodes of the BST without touching the first element of the linked list. It means that you need to sort each linked list starting from its second node. Then you will need to do an in-order traversal of this BST and print it to a new file called "file2.txt". This in-order traversal is a special traversal that prints the linked lists stored while visiting each node of the BST. You need to print one linked list per line to "file2.txt". Carefully observe the sample input and output displayed below to properly understand this. Please do the above tasks in a modular way as per the functions described below:

a) Create a function **readData**() that reads data from "file1.txt" and creates a BST of linked lists as described above. **20M**
b) Create a function **sortLists**() that uses any sorting algorithm of your choice to sort all the linked lists stored in the BST as per the above specifications. **17M**
c) Create a function **traverse**() that performs the in-order traversal as described above and prints the output to the file "file2.txt". Observe the sample output given below to understand it better. **17M**
d) Create a **main()** function that drives the above program in the order of the steps described above. **6M**

You are free to create your own structure definitions and choose the function signatures. Don't change the function names.

Sample Execution of the Code

If "file1.txt" consists of the following lines:

```
5
27 5 89 9 61
6 4 87 9 82 7 5
3 9 5 4 75 5 82 7
99 6 26 11 42 50 8
91 3 6 22 5 45 8 71 85
```

Your program execution would print something like this:

```
$ ./a.out file1.txt
```

"file2.txt" should contain the following lines:

```
3 4 5 5 7 9 75 82
6 4 5 7 9 82 87
27 5 9 61 89
91 3 5 6 8 22 45 71 85
99 6 8 11 26 42 50
```

The files given in the archive: "numbers_10.txt", "numbers_20.txt" and "numbers_50.txt" can be used as sample input files in place of "file1.txt" for your testing.

Submission Instructions

The directory **labtest_<yourIDNumber>** now contains "**LT1_Program_<yourIDNumber>.c**". Please upload this file on the portal as your submission. You can also make multiple submissions of the same file. After submitting your solution, please download it and verify if it has been correctly submitted or not. Repeat the submission if not correctly uploaded. Please get your submission verified by one of the invigilators before leaving the room.

=== End of document===