

Birla Institute of Technology and Science, Pilani,  
Second Semester 2023-24  
CS F211 – Data Structures and Algorithms  
Lab Test – 2 (Open Book)

=====

28/04/2024

Max. Marks: 60 M

Duration: 3hrs

=====

### 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\_Labtest2\_<yourIDNumber>** (Example: **DSA\_LabTest1\_2021A7PS1234P**) in your home directory. Create a file: **"LT2\_Program\_<yourIDNumber>.c"** in the above directory (Example: **LT2\_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 and create any number of new functions.

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

Assume you are the data analyst of a super market. Assume that there are 100 unique items that you are selling in the super market. Each item has an ID in the range [1..100]. Every customer purchases a few items, which are recorded as transactions in our supermarket billing software. A sample list of transactions is as follows:

```
1 5 7 100
74 65 62
64 94 100 35 1
67 87 94 100 45 67 23
89 46 73 85 90
83 74 82 2 5 94 10 32 25
67
68 56 93 92
94 89 72 40
98 94 68 94 100 57 48
97 89 94 56 72
100 99 98 97 6 5 94 76
87 94 92 98 90 99
64 28 47 95 10 75 38 36 100
73 49 48 20 19 17
18 17 16 15 15 94
86 84 83 82 81 80 89 98 96 95 94
66 76
94
98 64 27
```

Each row indicates a transaction. For example the first row indicates the transaction containing 4 items of IDs 1, 5, 7 and 100. Assume that each item occurs only once in transaction.

As a data analyst my task is to check which item is sold maximum. This has to be done in  $O(1)$  time only. I need to design an efficient storage technique of these transactions such that checking for maximum sold item has  $O(1)$  time complexity only.

- Create function **createData()** that creates a linked list of linked lists for the above transactions. Each transaction is a linked list and all the transactions are stored in another linked list. **This can be hard coded for the above dataset. Don't use file handling. 20M**
- Create a function **representData()** that takes the above linked list of linked lists as input parameter and creates an appropriate data representation to enable the operation described above. **30M**
- Create a function **getMax()** that takes the data structure created in part b and outputs the item that was sold the most in  $O(1)$  time. **5M**
- Create a **main()** function that drives the above program in the order of the steps described above. **5M**

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

## Submission Instructions

Upload **"LT12\_Program\_<yourIDNumber>.c"** 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===