Birla Institute of Technology & Science, Pilani
2nd Semester 2016-17 - CS F211 – Data Structures and Algorithms

Lab 4 (Evaluation 1) : 7th Feb 2017

Time: 170 minutes **Marks: 12 + 18 = 30**

Instructions:

- *This test consists of two problems (Problem 1 and Problem 2) specified in two different files.*
- All input expressions should be read from stdin (scanf) and output should be printed on stdout (printf).
- For first 150 minutes, only a subset of test cases will be visible to students after submitting the code on the portal. Only in last 20 minutes, all test cases will be made visible.
- At the end of 170 minute period, the online system will stop evaluating the submissions but it will accept it for additional 10 minutes. At the end of 180 minute period, it will stop accepting the submissions.
- Only the last submission by the student for each problem will be considered for evaluation, irrespective of earlier correct submission.
- Assuming that a problem contains M marks, in case of (Run-error/Compiler-error/Timelimit-error), evaluation will be done for M/2 marks only.
- Total marks of each problem contains some marks for modularity and proper structuring of code.
- All submitted source code will be later checked manually by the instructor and final marks will be awarded. Any case of plagiarism and/or hard coding of test cases will fetch 0 marks for the problem/evaluation component.
- Make sure to return 0 from the main() function in case of normal termination.

Problem 2 of 2

Expected Time: 100 minutes **Marks: 18**

Problem Statement

A. Write a procedure *mergeFIFO* to merge two sorted lists that are stored as FIFO lists *ls1* and *ls2* into a new sorted list *ls3* stored as a FIFO list using only the operations provided by ADT FIFO list in Problem 1.

B. Also, write a procedure *mergeLIST* to merge two sorted lists that are stored as linked lists (with first and last pointers) *ls1* and *ls2* into a new sorted list stored as a linked list (with first and last pointers) *ls3*. Note that *ls1* and *ls2* are not required hereafter and therefore nodes of *ls1* and *ls2* should be reused in *ls3* i.e. new nodes should not be created for *ls3.*

Whenever create is called (i.e. 0 key value is given as input), a new list should be created (say *ls1*, *ls2,* and so on).

Add following function calls to solution of Problem 1 (memProf has been defined in Problem 1):

Key	**Function to call**	**Format**	**Description**
5	mergeFIFO	5	Assuming the given two lists before this call (*ls1 and ls2)* are sorted, create a new sorted list (*ls3*), and call traverse function on *ls3*.
6	mergeLIST	6	Assuming the given two lists before this call (*ls1 and ls2)* are sorted, create a sorted list (*ls3*) by using the nodes of *ls1* and *ls2* and call traverse function on *ls3*.
7	memProf	7	Print *curHeapSize* and *maxHeapSize* variables on a line, separated by a tab.
-1		-1	stop the program.

Test Case 1:

Input	**Output**
0 5 19 23 45 56 87	19 23 45 56 87
0 4 -165 -98 -2 50	-165 -98 -2 50
5	-165 -98 -2 19 23 45 50 56 87
-1	

Test Case 2:

Input	**Output**
0 5 19 23 45 56 87	19 23 45 56 87
0 4 -165 -98 -2 50	-165 -98 -2 50
6	-165 -98 -2 19 23 45 50 56 87
-1	

Test Case 3:

Input	**Output**
0 5 19 23 45 56 87	19 23 45 56 87
7	50 50 (Note: these memory values are only notional)
0 4 -165 -98 -2 50	-165 -98 -2 50
7	90 90 (Note: these memory values are only notional)
6	-165 -98 -2 19 23 45 50 56 87
7	100 100 (Note: these memory values are only notional)
-1	