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

---

Lab 4 (Evaluation 1) : 11th Feb 2017

Time: 170 minutes **Marks: 8 + 22 = 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: 22**

Problem Statement

Insertion sort can be applied on a linked list without allocating any new space for intermediate nodes during sorting.

Implement following functions for “linked list of linked lists of characters”, as implemented in Problem 1, according to given input format:

4. Function **sortCharList**:
   - Key: 4
   - Format: 4 K

- Description:
  - "K" represents position of this CharList in WordList (starting with 1).
  - Implement insertion sort to sort K^{th} CharList in place i.e. after sorting original WordList should point to sorted K^{th} CharList at its K^{th} position.
  - Call printCharList after sorting the K^{th} CharList

5. Function **sortAllCharLists**:
   - Key: 5
   - Format: 5
   - Description:
     - Call sortCharList on all CharLists of the WordList
     - Call printWordList in the end

6. Function **sortWordList**:
   - Key: 6
   - Format: 6
   - Description: Apply in place insertion sort on WordList such that, after sorting, CharList C1 (c11, c12, …, c1N) appears before CharList C2 (c21, c22,…, c2M), if and only if:
     - c1j == c2j , for 1 <= j < i , and
       - either, c1i < c2i , OR, c1i doesn't exist
     - Call printWordList in the end

Note: These functions can be called in any order. All operations should be in place, on the same WordList.

Test Case 1:

Input	Output
1 5 some stories are good and 4 4 4 3 5 6 -1	d g o o a e r e m o s e i o r s s t a e r d g o o a d n a d n a e r d g o o e i o r s s t e m o s