CSE 3010 Data Structures & Algorithms
I Semester 2019-2020
Assignment 3

Date of Assignment: 20 February 2020 12:00 noon
Deadline: 22 February 2020 5:00 pm ***Points***: 10

Instructions:

1. Copying of answers from your friends or from the Internet is strictly prohibited and will not be tolerated at any cost.
2. You should turn in your own answers. If copying of any kind is found, penalties as per Code of Student Conduct will apply.
3. Read the instructions carefully. Write a program in C to achieve the task.
4. Viva voce will be conducted to understand how you solved the problem. If you fail to explain the solution during the viva voce, ZERO points will be awarded for this assignment.

Objective of Assignment: To be able to implement a sorting technique in C based on the description of the sorting technique.

An input dataset contains unsorted items. Each item in the input dataset consists of the following attributes:

- Employee number – Is an integer
- Employee name – Is a string
- Job code – Is a string
- Date of joining – Is a string

You are required to do the following:

1. Read the data from the text file `employees_data.txt`. The text file is given to you.
2. Take input from the user:
   - 1 if the list should be sorted on employee number
   - 2 if the list should be sorted on data of joining

   Any value outside of the range 1…2 is an invalid input. Your program should test for valid input.
3. Sort the list in ascending order using the sorting algorithm described below.

   Every odd-numbered iteration over the list should move the item with the largest value to the end of the list. Every even-numbered iteration should move the smallest item to the beginning of the list. Iteration refers to a complete pass through the list.

   See an example given below to illustrate the algorithm.

   Input dataset: 100, 520, 312, 21, 70, 0

1st odd-numbered iteration	[all remaining items in the list], 520
1st even-numbered iteration	0, [all remaining items in the list], 520
2nd odd-number iteration	0, [all remaining items in the list], 312, 520
2nd even-numbered iteration	0, 21, [all remaining items in the list], 312, 520
3rd odd-numbered iteration	0, 21, [remaining item in the list], 100, 312, 520
3rd even-numbered iteration	0, 21, 70, 100, 312, 520

   Note: In the example given above, 6 items are in the input dataset. For the list to be sorted, 6 iterations are were on the list. It is possible that the list gets sorted in an earlier iteration. The algorithm should be intelligent enough to know when the list is sorted and should not go through the remaining iterations.

4. Display the item as it moves to its correct position in each iteration, using the format given below.

```
520 has moved to the 6th position
0 has moved to the 1st position
312 has moved to the 5th position
21 has moved to the 2nd position
100 has moved to the 4th position
70 has moved to the 3rd position
```

Note: Since the items are stored in a linked list, the first item is stored in position 1 and the last item in position n. There is no need to start at 0 and end at n-1.

5. Display the sorted list of employees in the format given below. The list is sorted on `Date of Joining`. The format represents only a sample of the employee data. Your program should print the complete list.

#	Employee Number	Employee Name	Job Code	Date of Joining
1	14702	EMP_344	JC4_03	2010-01-01
2	33666	EMP_187	JC1_00	2010-01-08
3	15849	EMP_973	JC4_00	2010-01-09
4	15595	EMP_860	JC4_01	2010-01-11
5	15958	EMP_985	JC4_80	2010-01-11
6	407	EssMP_017	JC4_02	2010-01-16
7	15715	EMP_915	JC1_01	2010-01-19
8	15370	EMP_739	JC4_02	2010-01-20
9	15865	EMP_984	JC5_47	2010-01-20
10	14568	EMP_263	JC4_02	2010-01-22

Some important requirements

a. You can use only doubly linked list to implement the solution.
b. You may delete any of the doubly linked list operations that are not being used by this program.
c. You may change the implementation of the doubly link list operations, as you deem fit, without making it into a singly linked list☺.
d. You cannot use additional space to sort the list. It has to be in-place sorting.
e. You cannot use any other additional structures for any other purpose of the algorithm.
f. You can upload only a single file with the extension .zip. Please do not upload any All files including the header file and the data file should be zipped using your three-digit roll number for uploading. In case of failure to upload the relevant files required to execute your program, 5 points will be deducted.
g. You should ensure that your program compiles correctly. Programs having compilation errors will fetch ZERO points.

Seek clarifications if you do not understand anything from this description.