CSE 1320-002 **Assignment 10** **Summer 2018**

The assignment will be graded out of 100 points.

Due: Thursday, August 9, 2018 11:59pm.

Submission Guidelines:

Assignment will be submitted via Blackboard. If there are multiple files that you are submitting then you should zip your files into a single file and submit it. The name of this file should be in following format: lastname_firstname_UTANetID. If you are submitting a single file, make sure it is in the following format: word, txt or pdf and it should be named as mentioned above.

Make sure your name and your student ID are listed in your assignments.

If your assignment is not completed by the deadline, send it anyway and review it with the TA for partial credit. Do not take a zero or excessive late penalties just because it isn't working yet. We will make an effort to grade you on the work you have done.

Note: Please do not forget to zip all the files in ONE folder

Example:

Task1.c

After that, copy all the C files into one folder and compress(zip) the folder. Then, submit the compressed (zipped) folder on Blackboard.

Assignment Specification:

Please get the values from user as input.

Task 1 (100 pts.)

In a file called task1.c, write a menu based C program to maintain the **Employees'** records. Your program should take the following inputs:

1. Employee **first name** *(max. 25 characters)*
2. Employee **last name** (max. 25 characters)
3. Employee **PhoneNo** *(10 characters. e.g. 1112223344)*
4. Employee **Salary** *(int, e.g. 100000)*

You need to create a structure **Employee** with above information. Then, create a linked list where each list node contains an **Employee**. Your program should be able to take records of a **minimum of 5 employees**. After taking the records, you should provide 7 functionalities to the user.

1. Print records – prints records of all employees.

2. Add a new record– take a new record from the user for a new employee. Create a node for the new employee and add it to the **beginning** of the list.

3. Delete record(s) – to delete a record, ask for the last name of the employee from the user. If there are multiple employees with same last name, you must delete all of their records (you have to delete corresponding list nodes).

4. Search by PhoneNo – prints record of the employee with a given PhoneNo.

5. Search by Salary range – take two Salary max, min; then print records of all employees who have salary between [max, min] (inclusive).

6. Find the median salary – compute the median salary and print it. Also, print how many employees are above this median salary (do not forget to sort the list to compute median).

7. Exit the program–terminate on a specific input from the user. Let that specific input be an integer of value 0.

You should print the record in the following format:

```
First Name: firstname1, Last Name: lastname1, PhoneNo.: phoneno_1, Salary: salary_1
First Name: firstname2, Last Name: lastname2, PhoneNo.: phoneno_2, Salary: salary_2
…
First Name: firstname5, Last Name: lastname5, PhoneNo.: phoneno_5, Salary: salary_5
```

You should write each functionality from 1-7 in separate functions. You should provide a menu to the user as following:

For inputs:

Please indicate number of records you want to enter (min 5):
# of records After user gives the number of records, you should inform the user how to enter the records: Please input records of employees (enter a new line after each record), with following format: first name last name phoneNo salary

After user gives the inputs for the records, inform the user about the functionalities:
Print records (press 1)
Add a new record (press 2)
Delete record(s) (press 3)
Search by phoneNo (press 4)
Search by salary range (press 5)
Find median salary (press 7)
Exit the program (press 0)

After user chooses a functionality, your program performs that and provides this menu again to select another functionality. This goes on until user presses 0.

Instruction:
Do not forget to use
- malloc/calloc for a new node.
- free to delete it.

Assignment Guidelines:

There will be several programming assignments in this course, typically assigned on a weekly basis. All assignments will have equal weight. No assignment scores will be dropped. The following class policies regarding assignments will be followed:

- All assignments must be submitted via Blackboard.
- No extra credit will be provided.
- If you make multiple submissions to Blackboard for the same assignment, only the latest submission will be graded.

Late submission policy:

- All assignments are graded out of 100 points. Assignments submitted late will be penalized, at a rate of 4 penalty points per hour. The submission time will be the time shown on Blackboard. Any assignment submitted more than 25 hours late will receive no credit.
- Exceptions to late submission penalties will only be made for emergencies documented in writing, in strict adherence to UTA policy. For all such exception requests, the student must demonstrate that he or she made all efforts to notify the instructor as early as possible.
- Computer crashes, network crashes, and software or hardware failure will NOT be accepted as justification for late submissions. If you want to minimize chances of a late submission, aim to submit early. You can always revise your submission till the deadline.
- Sometimes students submit the wrong files on Blackboard. Unfortunately, no credit or waiver of late penalties can be provided in such cases.
- If you find yourself in an emergency situation and cannot deliver homework on time, immediately inform the instructor and teaching assistant. Even if you have a valid reason for delivering late an assignment, you must make a convincing case that you have notified the instructor and teaching assistant as early as possible.

If you want to minimize chances of a late submission, aim to submit early. You can always revise your submission till the deadline (maximum 3 attempts).