60-141
Intro to Algorithms & Programming II Winter 2014

Lab #11: Dynamic Data Structures
(Due at the end of the lab period)

Objective: In this Lab you will practice implementing and using dynamically allocated data structures.

Background:.
The problem of not having enough space to store data, or wasting spaces by having too much of it using a any static data structure, like array, can easily be overcome by using dynamic data structures that can grow or shrink in size at run time.

A linked-list is a simple dynamic data structure that is based on self-referential C structures. In this lab you will use such a self-referential structure Student as defined below, to implement and manipulate a simple linked-list called SLIST.

```
struct student {
      int ID;
      char name[40];
      struct student *next;
}
typedef struct student Student;
```

Work to do:
You are going to define the following functions:

1. **`Student *addToList(Student *SLIST);`**
   - "addToList" will ask for the ID and name of a new student from the user, dynamically create a new Student structure, assign the values for the members, add the new student structure at the END of the student list "SLIST" and return the modified list.

2. **`void printList(Student *SLIST);`**
   - "printList" will traverse through the list from the beginning to the end and will print info for each student in the format "ID Name\n".

3. **`void printListRR(Student *SLIST);`**
   - "printListRR" will print the list, RECURSIVELY, starting from the END of the list, in the format "ID Name\n".

4. **`void searchList(Student *SLIST);`**
   - "searchList" will ask the user to enter a student's ID, search the list for the student having that ID and will print the data in the format "ID Name\n". If failed, it will print "ID "ID" not found".

For your convenience, a partial program (Lab11.c) with the function prototypes and the main() function with a menu option is given. Document the whole program and define the necessary other functions.

EVALUATION:
You need to show your instructor the complete programs at the end of this lab, or at the beginning of your next lab. The marks you will receive for this lab are made of two parts: Lab work marks 8 and **attendance marks 2**. **Total 10 marks**.

Lab Work Mark: You will be evaluated based on your solutions for the problems based on the following scheme:

0 mark = No work done.
2 mark = Incomplete code / does not compile, with no/invalid documentation
4 marks = Complete running program with no/invalid documentation
6 marks = Incomplete code / does not compile, with proper documentation
8 marks = Complete running program with proper documentation

IMPORTANT:
ASK QUESTIONS IF YOU GET STUCK, BUT DO YOUR OWN CODE. ANY CODE SUSPECTED TO BE SIMILAR TO ANOTHER SUBMISSION WILL CAUSE BOTH SUBMISSIONS TO RECEIVE A ZERO MARK ON ALL LABS AND BE REPORTED FOR PLAGIARISM