CSC 215
Procedural Programming with C
Lab #11
Linked List

Lab Section

- Write the following structs:
  - Create the structure "Node" that contains the following members:
    - data: a char pointer (char*).
    - next: a pointer to the structure "Node".
  - Create the structure "Linkedlist" that contains the following members:
    - head: a pointer to structure Node.

- Write a program that does the following in the main function:
  - Your program should read arguments.
  - Declare a linked list LL. Don't forget to initialize the head of the Linked List.
  - Insert the arguments that start with an uppercase letter at the beginning of the Linked List in the order they arrived in. Use ***insertAtBeginning***.
  - Insert the arguments that start with a lowercase letter at the end of the Linked List in the order they arrived in. Use ***insertAtTheEnd***.
  - Before inserting, check if the word exist in the Linked List using ***findNode***. if it exist, delete it using ***deleteNode***.
  - Print the linked list data using ***displayLinkedList***.

- Write the following functions:
  - Write the function ***insertAtBeginning*** that takes a pointer to structure Linked List LL and a string ele. Allocates a node using ***malloc*** or ***calloc***, then inserts it into the beginning of the linked list. As for the data, get the size of ele using ***strlen***, allocate a string, and then copy the string to it using ***strcpy***.
    - `void insertAtBeginning(struct Linkedlist* LL, char ele[]);`

  - Write the function ***insertAtTheEnd*** that takes a pointer to structure Linked List LL and a string ele. Allocates a node using ***malloc*** or ***calloc***, then inserts it at the end of the linked list. As for the data, get the size of ele using ***strlen***, allocate a string, and then copy the string to it using ***strcpy***.
    - `void insertAtTheEnd(struct Linkedlist* LL, char ele[]);`

  - Write the function ***deleteNode*** that takes a pointer to structure Linked List LL and a string ele. Then removes the node with the sting in its data from the linked list and frees its space. Don't forget to free the string's space too.
    - `void deleteNode(struct Linkedlist* LL, char ele[]);`

  - Write the function ***findNode*** that takes a structure Linked List LL and string ele. Then return 1 if ele exist. Otherwise, return 0.
    - `int findNode(struct Linkedlist LL, char ele[]);`

  - Write the function ***displayLinkedList*** that takes a structure Linked List LL. Then prints all the linked list's nodes data separated by commas.
    - `void displayLinkedList(struct Linkedlist LL);`

- Show your program to the instructor. Then upload it to LMS under Lab11 Homework.

Example runs:

```
$ ./lab11
ERROR: The program must read at least an argument.
$ ./lab11 my name is Marwan and my car is White
The list:-
White, Marwan, name, and, car
```

SUBMIT POLICY: -

- Use the follow naming convention: Lab11_ID_FirstName_LastName.c
  - **Example:** Lab11_123456789_Marwan_Almaymoni.c
- Use a comment to write your name and ID at the beginning of the code.
- The Deadline is: 13/05/2015 right before the Lab starts.
- Late submissions will not be accepted.
- Email submissions will not be accepted.
- **-1 Point** for not following the naming convention.
- **-1 Point** for not writing your name and ID in the code inside a comment.
- **-8 Points** if the submitted program didn't work due to syntax errors.
- **-10 Points** for cheating and helping others cheat.