**CSCI 301 Moustafa Elsayed**

**Computer Science 2**

**Project 4: EXERCISING A DOUBLY-LINKED LIST CLASS**

**Introduction**

Doubly Linked list are useful when trying to access data in any order the user wants. In this program, the user enter a line of character with “#” in between them and with every “#” that is input the character entered before it will be removed from the list, until the user enters a “.”. Then the program will print the new list that has all the characters before a “#” removed.

**Data Structures**

This program uses 4 data structures:

* DList “list0” : This is the doubly linked list that is used to hold and organize the characters that the user inputs.
* A character type “ch” in the main file: This is used to hold the characters that the user inputs.
* An int “I” in the main file: This is used to hold the index of the characters entered by the user.
* A Boolean “flag” in the main file: This is used to flag the while loop when to stop

**Functions**

This program uses 4 functions in the source file:

* Void append(): This function checks if the first node is null. If it is null then the function creates the first node and adds the character in it. If not then the function accesses the last node and creates a new node then adds the character in it.
* Void remove\_last(): This function removes the last node added to the doubly linked list.
* Bool empty(): This function checks if the doubly linked list is empty
* Node\* get\_node(): This function creates the new node and creates the proper links of it.

**Main function**

The functions asks the user for a line of characters then opens a loop that will stop only if the user enters a “.”. In the loop the function checks if the character input is “#” then the function will call the function remove\_last(). If not then the function will add that character to the list. Then the function checks if the input is “.” Then the variable “flag” will change to True and the loop will stop. Finally the function will print the characters left in the list.