

Data Structures CE233

Alexandria University

Faculty of Engineering
SSP- Program.
Summer Course



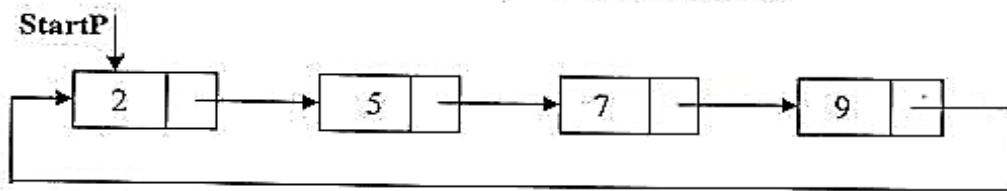
Assignment (2)

Assigned: 21/7/2014

Due: 27/7/2014

Prof. Dr. Magdy A. Ahmed

- 1) Write an algorithm to count the number of nodes in a singly linked list without head.
- 2) Write an algorithm to concatenate two singly linked list together.
- 3) Write an algorithm to invert a singly linked list.
- 4) Given a circular linked list of integers as shown below:



- (a) Write an algorithm to search for an integer , **num** , in a circularly linked list. The algorithm should return 1 if **num** is in the list and 0 otherwise.
 - (b) Write an algorithm that delete a node containing a target value , **num** , from a circularly linked list.
- 5) Let **List1** = (X_1, X_2, \dots, X_n) and **List2** = (Y_1, Y_2, \dots, Y_n) be two linked lists of integers. Assume that in each list, the nodes are in ascending order of their data field values. Write an algorithm to merge the two lists together to obtain a new linked list **List3** in which the nodes are also in this order. Following the merge, **List1** and **List2** do not exist as individual each node initially in **List1** and **List2** is now in **List3**. No additional nodes may be used.

Programming Assignment #2

Write a C Program(s) to implement the merge algorithm described in problem # 5.

- Use the insertion algorithm to create List1 and List2.
- Use the display algorithm to print out the merged list.
- Test your program using the following data :
List1 = (5 , 9 , 15 , 20 , 32 , 50 , 65 , 80 , 85 , 90)
List2 = (6 , 8 , 12 , 25 , 30 , 55 , 60 , 75 , 86 , 95 , 105 , 120 , 150 , 180)