**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA**

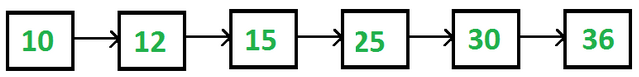
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Tutorial – 6 (Linked List)**

Course: B. Tech Semester: III

Course Code &Name: **18B11CI311 – Data Structures**

1. **Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?**   
   i) Insertion at the front of the linked list  
   ii) Insertion at the end of the linked list  
   iii) Deletion of the front node of the linked list  
   iv) Deletion of the end node of the linked list
2. Write an algorithm to insert an element at the second position in the linked list?
3. A variant of the linked list in which none of the node contains NULL pointer is?
4. Consider the following linked list



And following linked list representation

struct node {

int data;

struct node \*next;

}\*start ;

What will be printed by following statement?(Assume start is pointing to first node).

**Printf(“%d”, start->next->next->next->data);**

1. In doubly linked list each node has three fields, two fields for storing addresses and one for storing information of type **long double**. What will be the structure of such a node?
2. Let assume one doubly linked list of 5 nodes of type defined in question 5 then how much memory will be required for this linked list. What will be the PREV field of first node and NEXT field of last node in this linked list?

**Solution of Q1, Q2, Q3, Q4, Q5, Q6 and Q8 are available in below recording.**

[**https://drive.google.com/file/d/1-NXHKPyqiKlF9FYuuXWcuDcg6pz7zZ7v/view?usp=sharing**](https://drive.google.com/file/d/1-NXHKPyqiKlF9FYuuXWcuDcg6pz7zZ7v/view?usp=sharing)

1. Write an efficient algorithm to find out kth node from the end of the linked list.

**Algo Find\_kthNode\_From\_End(START,k)**

**i=1**

**Node\* temp1=START;**

**Node\* temp2=START;**

**While (temp1!=NULL)**

**If(i>k)**

**temp2=temp2->NEXT**

**Temp1=temp1->NEXT**

**i=i+1**

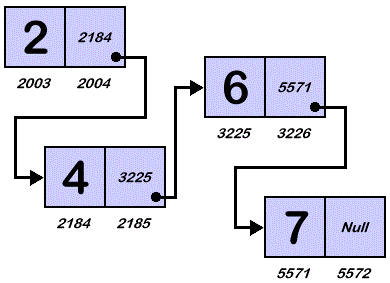
**if i<=k**

**print “size of linked list is less than %d”,k**

**else**

**print temp2->INFO**





If start is pointing to the first node of the linked list then consider the following statements

**Start=start->next**

**temp=start->next**

**Current=temp->next**

**What will be the value of address field of temp and data field of current? ……… …….…**

1. We are given a pointer to the first element of a linked list L. There are two possibilities for L, it either ends (snake) or its last element points back to one of the earlier elements in the list (snail). Give an algorithm that tests whether a given list L is a snake or a snail.

<https://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/>

1. **Consider sorted singly linked list having following nodes**

**10->30->50->70->NULL**

You are given pointer to node 50 and a new node having value 40. Can you insert node 40 correctly in the list maintaining the ascending order?

**Solution:**

**Explanation: Insert new node after 50 with data value 50, then put 40 in data field of node 50.**

**Algorithm Insert(Temp)// temp is a pointer to node 50**

**Node \*New\_node;**

**New\_node=allocate memory**

**New\_node->INFO=50**

**Temp->INFO=40 // replace 50 by 40**

**Temp->NEXT=New\_node // insert new node after node 50**

1. Write an algorithm to delete a node from given location in doubly linked list.

**Refer slides and lecture recording.**