Assignment- 03

Investigating different architectures of linked list [CO 3]

| Deadline:

Instructions:
For every task you need to show tracing/simulation, codes and final output. If tracing/simulation is missing, half of the marks will be deducted. Try to maintain sequence. Write name, student id, assignment number and date of submission clearly.

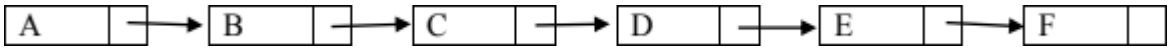
Each of the first 10 tasks should have 3 parts-

I. **Pseudo code/ java code**
II. **Tracing**
III. **Final output**

Task 1

Create a Linked List of 6 elements using your nickname.

Hint: If your name is John, then the elements should be –



Where,

X = Last 4 digit of your id %4 + Last 3 digit of your id %6

Y = Last 4 digit of your id %5 + Last 3 digit of your id % 9

A = (X+1)_John_(Y+10)

B = (X+2)_John_(Y+20)

C = (X+3)_John_(Y+30)

D = (X+4)_John_(Y+40)

E = (X+5)_John_(Y+50)

F = (X+6)_John_(Y+60)

Task 2

Print the elements in the List.

Task 3

Count the number of elements in the List.

Task 4

Get an element from the List where an index is given. Index will be (last 3 digit of your ID%5).

Task 5

Set an element in the List where an index is given. Index will be (last 3 digit of your ID%4).

Task 6

Search an element from the List where an index is given where index will be (last 3 digit of your ID%3).

Task 7

Insert an element in the List in –

I. First position
II. Last position
III. A random position where Index will be (last 3 digit of your ID%5)

Task 8

Remove an element from the list from –

I. First position
II. Last position
III. A random position where Index will be (last 3 digit of your ID%4)

Task 9

Make a reversed copy of the List.

Task 10

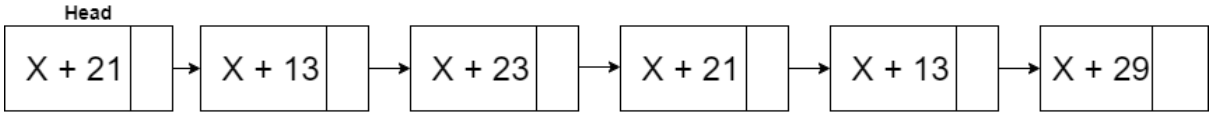
Rotate the List –

I. To the right by 1 position.
II. To the left by 1 position.

Task 11

X = Last 3 Digit of your id % 25 + Last 3 digit of your id % 30 + 2

Consider the following structure of singly linked list,



a) **Write a code** that sorts the linked list and **show output/simulation**.

Then, consider the following code

```
// Remove duplicates from a sorted list
Public class LinkedList {
   Node head;
public void RemoveDuplicates() {
        node current = head;
   while (current != null) {
         Node temp = current;
   while(temp!=null && temp.element==current.element) {
            temp = temp.next;
         }
         current.next = temp;
         current = current.next;
      }
   }
}
```

```
//Attributes of Node class
 public  class Node
   {
      int element;
      Node next;
   }
```

b) **Use the sorted linked list** found in Part - a, and **simulate each step** using the RemoveDuplicate method

c) **Show the resulting list after calling the method.**