**Assignment 4**

**Question 1**

Implement deletion operation from the end of the linked list and Insertion operation from the

beginning of the linked list.

**Answer 1**

Insertion in the beginning:

*def InsertBeg(self, new\_data):*

*new\_node = Node(new\_data)*

*new\_node.next = self.head*

*self.head = new\_node*

Deletion at the end:

*def removeLastNode(head):*

*if head == None:*

*return None*

*if head.next == None:*

*head = None*

*return None*

*second\_last = head*

*while(second\_last.next.next):*

*second\_last = second\_last.next*

*second\_last.next = None*

*return head*

**Question 2**

Implement binary search using python language.

(Write a function which returns the index of x in given array arr if present, else returns -1)

**Answer 2**

*def binary\_search(arr,element):*

*low = 0*

*high = len(arr)-1*

*if\_found = False*

*while( low<=high and not if\_found):*

*mid = (low + high)//2*

*if arr[mid] == element :*

*if\_found = True*

*else:*

*if element < arr[mid]:*

*high = mid - 1*

*else:*

*low = mid + 1*

*if if\_found==True:*

*return("Element {} found at index {} ".format(element,mid))*

*else:*

*return("-1")*

**Question 3**

Write a Python program to find the middle of a linked list.

**Answer 3**

*class Node:*

*def \_\_init\_\_(self, value):*

*self.data = value*

*self.next = None*

*class LinkedList:*

*def \_\_init\_\_(self):*

*self.head = None*

*def push(self, new\_data):*

*new\_node = Node(new\_data)*

*new\_node.next = self.head*

*self.head = new\_node*

*def printMiddle(self):*

*temp = self.head*

*count = 0*

*while self.head:*

*if (count & 1):*

*temp = temp.next*

*self.head = self.head.next*

*count += 1*

*print(temp.data)*