Assignment 4

A1) Deletion

```
def deleteBeg(self):
  if self.head == None:
  Print ("LL is empty!")
   return
  self.head = self.head.next
```

Insertion

```
def insertEnd(self, data):
 new = Node(data)
 if self.head == None:
  self.head= new
  return
 temp = self.head
 while temp.next != None:
 temp = temp.next
  temp.next = new
```

A2)

```
def binary_search(arr, low, high, x):
    if high >= low:
        mid = (high + low) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binary_search(arr, low, mid-1, x)
        else:
            return binary_search(arr, mid+1, high, x)
    else
        return -1

arr = [2, 3, 4, 10, 40]
x = 10
result = binary_search(arr, 0, len(arr)-1, x)
if result != -1:
    print("Element is present at index", str(result))
else:
    print("Element is not present in array")
```

A3)

```
class Node:
    def __init__(self, data):
        self.data = data
        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):
        slow_ptr = self.head
        fast_ptr = self.head
        if self.head is not None:
            while (fast_ptr is not None and fast_ptr.next
                   is not None):
                fast_ptr = fast_ptr.next.next
                slow_ptr = slow_ptr.next
            print("The middle element is:", slow_ptr.data)
```