Implementation of Doubly linked list (Traversing the Nodes, searching for a Node, Appending Nodes, Deleting Nodes):

```
class Node:
    def __init__(self, data = None):
        self.data = data
        self.next = None
        self.prev = None
class DoublyLinkedList:
    def __init__(self):
        self.first = None
    def insertAtEnd(self, data):
        temp = Node(data)
        if(self.first == None):
            self.first=temp
        else:
            cur = self.first
            while(cur.next != None):
                cur = cur.next
            cur.next = temp
            temp.prev = cur
    def deleteFirst(self):
        if(self.first== None):
            print("list is empty")
        elif(self.first.next == None):
            print("the deleted item is",self.first.data)
            self.first = None
        else:
            cur=self.first
            self.first=self.first.next
            self.first.prev = None
            print("the deleted item is",cur.data)
    def display(self):
        if(self.first== None):
            print("list is empty")
            return
        cur = self.first
        while(cur):
            print(cur.data, end = " ")
            cur = cur.next
    def search(self,item):
        if(self.first== None):
            print("list is empty")
            return
        cur = self.first
        while cur != None:
            if cur.data == item:
                print("Item is present in the Linked list")
                return
            else:
```

```
            cur = cur.next
        print("Item is not present in the Linked list")
#Doubly Linked List
dll = DoublyLinkedList()
while(True):
    ch = int(input("\nEnter your choice 1-insert 2-delete 3-search 4-display 5-exit :"))
    if(ch == 1):
        item = int(input("Enter the element to insert:"))
        dll.insertAtEnd(item)
        dll.display()
    elif(ch == 2):
        dll. deleteFirst()
        dll.display()
    elif(ch == 3):
        item = int(input("Enter the element to search:"))
        dll.search(item)
    elif(ch == 4):
        dll.display()
    else:
        break
```

Implementation of Circular Singly linked list (Traversing the Nodes, searching for a Node, Appending Nodes and Deleting Nodes):

```
class Node:
    def __init__(self, data = None):
        self.data = data
        self.next = None
class CircularLinkedList:
    def __init__(self):
        self.first = None
    def insertAtEnd(self, data):
        temp = Node(data)
        if(self.first == None):
            self.first = temp
            self.first.next = temp
        else:
            cur = self.first
            while(cur.next != self.first):
                cur = cur.next
            cur.next = temp
            temp.next = self.first
    def deleteAtEnd(self):
        if(self.first== None):
            print("list is empty")
        elif(self.first.next == self.first):
            print("the deleted item is",self.first.data)
            self.first = None
        else:
            cur=self.first
            while(cur.next != self.first):
                pr = cur
                cur = cur.next
            pr.next = self.first
            print("the deleted item is",cur.data)
    def display(self):
        if(self.first== None):
            print("list is empty")
            return
        cur = self.first
        while(True):
            print(cur.data, end = " ")
            cur = cur.next
            if(cur == self.first):
                break
    def search(self,item):
        if(self.first== None):
            print("list is empty")
            return
        cur = self.first
        while (cur.next != self.first):
            if (cur.data == item):
                print("Item is present in the linked list")
                return
            else:
                cur = cur.next
        print("Item is not present in the linked list")
#Circular Linked List
cll = CircularLinkedList()
while(True):
    ch = int(input("\nEnter your choice 1-insert 2-delete 3-search 4-display 5-exit :"))
    if(ch == 1):
        item = input("Enter the element to insert:")
        cll.insertAtEnd(item)
        cll.display()
    elif(ch == 2):
        cll.deleteAtEnd()
        cll.display()
    elif(ch == 3):
        item = int(input("Enter the element to search:"))
        cll.search(item)
    elif(ch == 4):
        cll.display()
    else:
        break
```