```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def append(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
            self.last_node = new_node
        else:
            self.last_node.next = new_node
            self.last_node = new_node

    def prepend(self, data):
        new_node = Node(data)
        new_node.next = self.head
        self.head = new_node

    def insertend(self,data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
            self.last_node = new_node
        else:
```

```
            self.last_node.next = new_node
            self.last_node = new_node

    def delete_begin(self):
        if self.head:
            self.head = self.head.next

    def delete_end(self):
        if self.head:
            current = self.head
            previous = None
            while current.next:
                previous = current
                current = current.next
            if previous:
                previous.next = None
            else:
                self.head = None

    def display(self):
        temp = self.head
        while temp is not None:
            print(temp.data, "-->", end=" ")
            temp = temp.next
        print("None")
l = LinkedList()

n = int(input())
for i in range(n):
```

```
    data = int(input())
    l.append(data)

print("\nLinked List after appending:")
l.display()

k= int(input())
l.prepend(k)

print("\nLinked List after prepending:")
l.display()

m = int(input())
l.insertend(m)

print("\nAfter inserted at the end")
l.display()

l.delete_begin()

print("\n After delete at the begining")
l.display()

l.delete_end()

print("\n After delete at the end")
l.display()
```