

```
In [4]: class Node:
        def __init__(self, data):
            self.data = data
            self.next_node = None
            self.prev_node = None
```

```
In [5]: class LinkedList:
        def __init__(self):
            self.head = None

        def addNode(self, data):
            new_node = Node(data)

            if not self.head:
                self.head = new_node
                return

            current = self.head
            while current.next_node:
                current = current.next_node

            current.next_node = new_node

        def deleteNode(self, data):
            current = self.head

            if current and current.data == data:
                self.head = current.next_node
                return

            while current.next_node and current.next_node.data != data:
                current = current.next_node

            if current.next_node:
                current.next_node = current.next_node.next_node

        def printList(self):
            current = self.head
            while current:
                print(current.data, end=" ")
                current = current.next_node
            print()
```

```
In [6]: cpeList = LinkedList()
        cpeList.addNode("C")
        cpeList.addNode("P")
        cpeList.addNode("E")
        cpeList.printList()

        cpeList.deleteNode("P")
        cpeList.printList()
```

C P E
C E

```
In [7]: class UltimateLinkedList:
        def __init__(self):
            self.head = None
```

```

def addNode(self, data):
    new_node = Node(data)

    if not self.head:
        self.head = new_node
        new_node.next_node = new_node
        new_node.prev_node = new_node
    else:
        last_node = self.head.prev_node
        last_node.next_node = new_node
        new_node.prev_node = last_node
        new_node.next_node = self.head
        self.head.prev_node = new_node

def deleteNode(self, data):
    if not self.head:
        return

    current = self.head

    while True:
        if current.data == data:
            current.prev_node.next_node = current.next_node
            current.next_node.prev_node = current.prev_node
            if current == self.head:
                self.head = current.next_node
            return
        current = current.next_node
        if current == self.head:
            break

def printList(self):
    if not self.head:
        return

    current = self.head

    while True:
        print(current.data, end=" ")
        current = current.next_node
        if current == self.head:
            break
    print

```

```

In [8]: cpeList = UltimateLinkedList()
        cpeList.addNode("C")
        cpeList.addNode("P")
        cpeList.addNode("E")
        cpeList.printList()

```

C P E

```

In [126... cpeList.deleteNode("P")
            cpeList.printList()

```

C E