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

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

  def detectAndRemoveLoop(self):
    slow_p = fast_p = self.head

    while(slow_p and fast_p and fast_p.next):
      slow_p = slow_p.next
      fast_p = fast_p.next.next

      if slow_p == fast_p:
        self.removeLoop(slow_p)

        return True

    return False

  def removeLoop(self, loop_node):
    p1 = loop_node
    p2 = loop_node

    k = 1
    while(p1.next != p2):
      p1 = p1.next
      k += 1

    p1 = self.head

    p2 = self.head
    for i in range(k):
      p2 = p2.next

    while(p2 != p1):
      p1 = p1.next
      p2 = p2.next

    while(p2.next != p1):
      p2 = p2.next

    p2.next = None

  def push(self, new_data):
    new_node = Node(new_data)
    new_node.next = self.head
    self.head = new_node

  def printList(self):
    temp = self.head
    while(temp):
      print(temp.data, end = ' ')
      temp = temp.next
```

```
link_list = LinkedList()
link_list.push(10)
link_list.push(4)
link_list.push(15)
link_list.push(20)
link_list.push(50)

#Creating a loop
link_list.head.next.next.next.next.next = link_list.head.next.next

link_list.detectAndRemoveLoop()

print("\nLinked List after removing loop")
link_list.printList()
```

```

Linked List after removing loop
50 20 15 4 10
```

✓ 0s completed at 1:57 PM ● ×