Linked List

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

class LinkedList:

   def __init__(self):
      self.head=None
```

1.Search an element

```
def search_element(self, data):
      if self.head is None:
         print("List is Empty")
         return
      itr = self.head
      while itr:
         if itr.data == data:
            print("Element is present in list")
            return
         itr = itr.next
      print("Element is not present in the list")
```

2. Access an element at nth position

```
def get_element(self, position):
      if self.head is None:
         print("Linked List is Empty")
         return
      itr = self.head
      counter = 0
      while itr:
         counter+=1
         if counter == position:
            print("Element present at this positon is " + str(itr.data))
            return
         itr = itr.next
      print("No Element is present at this position")
```

3. Insertion at nth position

```
    def insert_at_position(self, data, pos):
        if self.head is None:
            print("Linked List is Empty")
            return
        itr = self.head
        counter = 0
        previousNode = None
        while itr:
            counter+=1
            if counter == pos:
                if previousNode is None:
                    self.insert_at_begining(data)
                    return
                if itr.next == None:
                    self.insert_at_end(data)
                    return
                node = Node(data,previousNode.next)
                previousNode.next = node
                return
            else:
                previousNode = itr
                itr = itr.next
        print("Invalid position")
```

4. Deletion at nth position

```
def deletion_at_position(self, pos):
        if self.head is None:
            print("Linked List is Empty")
            return
        itr = self.head
        previousNode = None
        counter = 0
        while itr:
            counter+=1
            if counter == pos:
                if previousNode is None:
                    if itr.next != None:
                        self.head = itr.next
                    else:
                        self.head = None
                else:
                    if itr.next == None:
                        previousNode.next = None
                    else:
                        previousNode.next = itr.next
                return
            else:
                previousNode = itr
                itr = itr.next
        print("Invalid Position")



if __name__=='__main__':
    llist = LinkedList()
    llist.insert_element([10,20,30,40,50,60,70,80,90,100])
    #llist.search_element(20)
    #llist.get_element(2)
    #llist.insert_at_position(25,1)
    llist.deletion_at_position(100)
    #llist.print()
```