# A simple Python program to introduce a linked list

# Node class

class Node:

# Function to initialise the node object

def \_\_init\_\_(self, data):

self.data = data # Assign data

self.next = None # Initialize next as null

# Linked List class contains a Node object

class LinkedList:

# Function to initialize head

def \_\_init\_\_(self):

self.head = None

def insertNode(self, newdata):

n=Node(newdata)

n.next=self.head

self.head=n

def insertAfter(self, newdata, givennode):

n=Node(newdata)

n.next=givennode.next

givennode.next=n

def insertEnd(self,newdata):

n=Node(newdata)

temp=self.head

while(temp.next):

temp=temp.next

temp.next=n

def deleteKey(self,key):

temp=self.head

if temp.data==key:

self.head=temp.next

else:

while(temp.next):

if temp.next.data==key:

temp.next=temp.next.next

temp=temp.next

def deletePosition(self,position):

temp=self.head

for i in range(1,position-1):

temp=temp.next

temp.next=temp.next.next

def lengthList(self):

temp=self.head

count=0

while(temp.next):

count+=1

temp=temp.next

print "length",count

def printList(self):

temp=self.head

while(temp.next):

print temp.data

temp=temp.next

print "new"

# Code execution starts here

# Start with the empty list

llist = LinkedList()

llist.head = Node(1)

second = Node(2)

third = Node(3)

llist.head.next = second; # Link first node with second

second.next = third; # Link second node with the third node

llist.printList()

llist.insertNode(4)

llist.printList()

llist.insertAfter(5,second)

llist.printList()

llist.insertEnd(6)

llist.printList()

llist.deleteKey(5)

llist.printList()

llist.deletePosition(3)

llist.printList()

llist.lengthList()