# stack implimentation using linklist  
class node:#node class that create link list  
 def \_\_init\_\_(self, data, next):#constructor to define object of node class  
 self.data = data  
 self.next = next  
  
def push(head, data):#method to push data in stack  
 new\_node = node(data, head)#create new node  
 head = new\_node  
 return head  
  
def pop(head):#method to pop data from stack  
 if head is None:  
 return None  
 data = head.data  
 print()  
 print(data,'is poppped')  
 head = head.next  
 return data, head  
  
def traverse(head):#method to traverse/print data of link list  
 while head is not None:  
 print(head.data)  
 head = head.next  
  
head = node(1, None) # Create a node with data 1  
print(head.data,'is pushed')  
head = push(head, 2) # Push 2 onto the stack  
print(head.data,'is pushed')  
head = push(head, 3) # Push 3 onto the stack  
print(head.data,'is pushed')  
head = push(head, 4) # Push 4 onto the stack  
print(head.data,'is pushed')  
  
data, head = pop(head) # Pop the stack  
print('\nprint stack using traverse method')  
traverse(head) # Print the stack