السؤال الأول

private int self;  
 private int data;  
def-int-(self,data):  
self.data = data  
self. next = None

السؤال ثاني

private T item;  
  
public void setItem(T item) {  
 this.item = item;  
}  
  
public T getItem() {  
 return item;  
  
}

السؤال الثالث

private int self;  
private int data;  
  
private int current;  
def\_init\_(self,data):  
class Singlylinkedlisr:  
def\_init\_(self,data):  
self.head = None  
self.tail = None  
  
def is\_empty(self):  
return self.head is None  
  
def append(self,data):  
  
current.next == self.tail:  
current.next.next;  
break;  
  
def search(self,data):  
current = self.head  
while current is not None:  
if current.data ==  
  
return True  
current = current.next  
 elements.append(current.data)  
current = current.next  
print(elements)

السؤال 4

ublic class Node {  
 private int self;  
 private int value;  
 def\_init\_(sehf,value):  
 self.value.value;  
 self.next = None;  
   
 class singlylinkedlist:  
 def\_init\_(self):  
 self.head =None   
 self.teil =None  
 self.length =0;  
   
 def isEmpty(self):  
 return self.length ==0  
 def size(self):  
 return self.length  
   
 def first(self):  
 if self.isEmpty():  
 return None  
 return.self.tail.value  
   
 def addfirst(self,value):  
 newNode = Node(value)  
 if self.isEmpty():  
 self.head = newNode  
 self.tail = newNode  
  
 else:  
 newNode.next =   
 self.head  
self.head = newNode  
self.length += 1  
  
def addlast(self,value):

السؤال خامس

public class singlylinkedlist {  
 private Node<T> head;  
 singlylinkedlist implementation:  
   
 public boolean equals(objectobj)  
 if(this == obj)  
 return true;  
 if(obj == null)  
   
   
   
 while (currentNode)

السؤال السادس

public class u6 {  
 def head is None or head.next  
 is None:  
 return "Second\_to\_last"  
 Node doesn exist.  
 current = head   
 prev head  
 while current.next.is not  
 None   
 current = current.next  
 prev =  
  
 public None getCurrent() {  
 return current;  
 return prev;  
 }  
}

السؤال 7

public class u7 {  
 def\_init\_(self,data):  
 self.data = data  
 self.next = None  
  
 class singlylinkedlist:  
 def\_init\_(self,data):  
 self.head = None  
  
 class singlylinkedlist:  
 def\_init\_(self,data):  
 self.head = None  
def add (self,data):  
 new\_node = Node (data)  
  
 if self.head is None:  
 else  
 current = self.head  
 while current.next is  
 not None  
  
 def size(self):  
  
 count = 0  
 current = self.head  
 while current is not None:  
 current = current.next  
 return count   
}

السؤال 8

public class u8 {  
 def-int-(self,data):  
 self.data = data  
 self.next = None  
 private int addlast;  
 class singlylinkedlist:  
 def\_init\_(self):  
 self.head = None  
 self.tail = None   
   
 linked\_list =singlylinkedlist()   
 linked\_list.addlast(1)  
 linked\_list addlast(2)  
 linked\_list addlast(3)  
 linked\_list addlast(4)  
 linked\_list addlast(5)  
   
 linked\_list.rotate()  
   
 current = linked\_list.head  
 while current:  
   
 print(current.data)  
 current = current.nxet

السؤال 9

1. قم بتهيئة قائمة مرتبطة فارغة جديدة..
2. قم بتعين المؤشر cur إلى العقدة الرئيسية .
3. اجتياز عقدا حتا تصل إلى النهاية لكل عقدة قم بأنشاء عقدة جديدة.
4. قم بالحاق بنهاية باستخدام عملية الارتباط المناسبة
5. اجتياز العقدة حتى تصل الى النهاية.
6. قم بإرجاع والذي يحتوي على جميع عقد متبوعة بجميع عقد.

public class u9 {  
 def\_init\_(self,val=0,next=None):  
 self.val.val  
 self.next = next  
   
 def rverselinkedlist(head):  
 previous = None  
 current head  
 next = None  
 while current is not None:  
 next = current.next  
 cirrent.next = previous  
 previous = current  
 current = next  
 head = previous  
 return head  
  
}

السؤال 10

public class u10 {  
 def reverse\_linked\_list(head)  
 previous = None  
 current = head   
 next = None  
 while current is not None:  
 next =current. next  
 Store the next node  
 current.next = previous  
 Reverse the pointer  
 previous = current  
   
 current =  
  
 public next getNode() {  
 return node;  
 head = previous  
 }  
}