01. Write a python program to remove punctuation from a string

punctuation = '''''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

my\_str = input("Enter a string: ")

no\_punct = ""

for char in my\_str:

if char not in punctuation:

no\_punct = no\_punct + char

print(no\_punct)

02. Write a python program to sort the elements of the circular linked list

class Node:

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

self.data = data;

self.next = None;

class CreateList:

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

self.head = Node(None);

self.tail = Node(None);

self.head.next = self.tail;

self.tail.next = self.head;

def add(self,data):

newNode = Node(data);

if self.head.data is None:

self.head = newNode;

self.tail = newNode;

newNode.next = self.head;

else:

self.tail.next = newNode;

self.tail = newNode;

self.tail.next = self.head;

def sortList(self):

current = self.head;

if(self.head == None):

print("List is empty");

else:

while(True):

index = current.next;

while(index != self.head):

if(current.data > index.data):

temp = current.data;

current.data = index.data;

index.data = temp;

index= index.next;

current =current.next;

if(current.next == self.head):

break;

def display(self):

current = self.head;

if self.head is None:

print("List is empty");

return;

else:

print(current.data, end= ' ');

while(current.next != self.head):

current = current.next;

print(current.data,end=' ');

print("\n");

class CircularLinkedList:

cl = CreateList();

cl.add(70);

cl.add(90);

cl.add(20);

cl.add(100);

cl.add(50);

print("Original list: ");

cl.display();

cl.sortList();

print("Sorted list: ");

cl.display();

03. Write a python program to search an element from a doubly linked list

class Node:

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

self.data = data;

self.previous = None;

self.next = None;

class SearchList:

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

self.head = None;

self.tail = None;

def addNode(self, data):

newNode = Node(data);

if(self.head == None):

self.head = self.tail = newNode;

self.head.previous = None;

self.tail.next = None;

else:

self.tail.next = newNode;

newNode.previous = self.tail;

self.tail = newNode;

self.tail.next = None;

def searchNode(self, value):

i = 1;

flag = False;

current = self.head;

if(self.head == None):

print("List is empty");

return;

while(current != None):

if(current.data == value):

flag = True;

break;

current = current.next;

i = i + 1;

if(flag):

print("Node is present in the list at the position : " + str(i));

else:

print("Node is not present in the list");

dList = SearchList();

dList.addNode(1);

dList.addNode(5);

dList.addNode(4);

dList.addNode(2);

dList.addNode(3);

dList.searchNode(4);

dList.searchNode(9);

04. Write a python program to print all pronic numbers from 1 to 100

def isPronicNumber(num):

flag = False;

for j in range(1, num+1):

if((j\*(j+1)) == num):

flag = True;

break;

return flag;

print("Pronic numbers between 1 and 100: ");

for i in range(1, 101):

if(isPronicNumber(i)):

print(i),

print(" "),

05. Write a python program to print the elements of array in reverse order

arr = [1, 2, 3, 4, 5];

print("Original array: ");

for i in range(0, len(arr)):

print(arr[i]),

print("Array in reverse order: ");

for i in range(len(arr)-1, -1, -1):

print(arr[i]),