Write a program in Java to insert a new element in a sorted circular linked list

Program:

class Node

{

int data;

Node next;

Node(int d)

{

data = d;

next = null;

}

}

class LinkedList

{

Node head;

// Constructor

LinkedList() { head = null; }

/\* function to insert a new\_node in a list in sorted way.

Note that this function expects a pointer to head node

as this can modify the head of the input linked list \*/

void sortedInsert(Node new\_node)

{

Node current = head;

// Case 1 of the above algo

if (current == null)

{

new\_node.next = new\_node;

head = new\_node;

}

// Case 2 of the above algo

else if (current.data >= new\_node.data)

{

/\* If value is smaller than head's value then

we need to change next of last node \*/

while (current.next != head)

current = current.next;

current.next = new\_node;

new\_node.next = head;

head = new\_node;

}

// Case 3 of the above algo

else

{

/\* Locate the node before the point of insertion \*/

while (current.next != head &&

current.next.data < new\_node.data)

current = current.next;

new\_node.next = current.next;

current.next = new\_node;

}

}

// Utility method to print a linked list

void printList()

{

if (head != null)

{

Node temp = head;

do

{

System.out.print(temp.data + " ");

temp = temp.next;

} while (temp != head);

}

}

// Driver code to test above

public static void main(String[] args)

{

LinkedList list = new LinkedList();

// Creating the linkedlist

int arr[] = new int[] {12, 56, 2, 11, 1, 90};

/\* start with empty linked list \*/

Node temp = null;

/\* Create linked list from the array arr[].

Created linked list will be 1->2->11->12->56->90\*/

for (int i = 0; i < 6; i++)

{

temp = new Node(arr[i]);

list.sortedInsert(temp);

}

list.printList();

}

}

Output:

1 2 11 12 56 90