**Question 1: Implement the bucket and Radix sort using linked list.**

**Code:**

***BucketSort:***

**LinkedList Class:**

class LinkedList

{

private Node start;

private int length;

public LinkedList()

{

start = new Node();

length = 0;

}

public void InsertAtEnd(int val)

{

if (start.next == null)

{

Node n = new Node(val);

start.next = n;

}

else

{

Node temp = start.next;

while (temp.next != null)

temp = temp.next;

temp.next = new Node(val);

}

}

public void Sort()

{

Node temp = start.next;

for (Node i = temp; i.GetNext() != null; i = i.GetNext())

{

Node small = i;

for (Node j = i.GetNext(); j != null; j = j.GetNext())

{

if (i.GetData() > j.GetData())

small = j;

}

int temp1 = i.GetData();

i.SetData(small.GetData());

small.SetData(temp1);

}

}

public void Display()

{

Console.Write("Head -> ");

Node curr = start;

while (curr.next != null)

{

curr = curr.next;

Console.Write(curr.data);

Console.Write(" -> ");

}

Console.WriteLine("Null");

}

}

**Program Class:**

static void Main(string[] args)

{

LinkedList buk1 = new LinkedList();

LinkedList buk2 = new LinkedList();

LinkedList buk3 = new LinkedList();

LinkedList buk4 = new LinkedList();

LinkedList buk5 = new LinkedList();

Console.Write("Enter how many elements you want to enter = ");

int ans = Convert.ToInt32(Console.ReadLine());

int[] arr = new int[ans];

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

{

Console.Write("Enter element " + (i + 1) + " in range of 0 to 100 = ");

arr[i] = Convert.ToInt32(Console.ReadLine());

}

int div = ans / 5;

int sort = 0;

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

{

if (arr[i] >= 0 && arr[i] < 20)

{

buk1.InsertAtEnd(arr[i]);

buk1.Sort();

}

else if (arr[i] >= 20 && arr[i] < 40)

{

buk2.InsertAtEnd(arr[i]);

buk2.Sort();

}

else if (arr[i] >= 40 && arr[i] < 60)

{

buk3.InsertAtEnd(arr[i]);

buk3.Sort();

}

else if (arr[i] >= 60 && arr[i] < 80)

{

buk4.InsertAtEnd(arr[i]);

buk4.Sort();

}

else if (arr[i] >= 80 && arr[i] < 100)

{

buk5.InsertAtEnd(arr[i]);

buk5.Sort();

}

}

buk1.Display();

buk2.Display();

buk3.Display();

buk4.Display();

buk5.Display();

}

**Node Class:**

class Node

{

public Node next;

public int data;

public Node(int d)

{

data = d;

next = null;

}

public Node()

{

next = null;

}

public Node(int n, Node o)

{

next = o;

data = n;

}

public Node GetNext()

{

return next;

}

public int GetData()

{

return data;

}

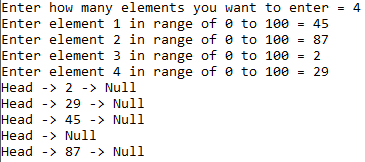
public void SetData(int val)

{

data = val;

}

**Output:**



***RadixSort:***

**Code:**

private int[] data;

private IList<IList<int>> digits = new List<IList<int>>();

private int maxLength = 0;

public Radix()

{

for (int i = 0; i < 10; i++)

{

digits.Add(new List<int>());

}

Console.Write("Enter the Length of an array : ");

int count = int.Parse(Console.ReadLine());

data = new int[count];

Console.ReadLine();

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

{

Console.Write("Enter number {0} : ", i + 1);

data[i] = int.Parse(Console.ReadLine());

if (maxLength < data[i].ToString().Length)

maxLength = data[i].ToString().Length;

}

}

public void RadixSort()

{

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

{

for (int j = 0; j < data.Length; j++)

{

int digit = (int)((data[j] % Math.Pow(10, i + 1)) / Math.Pow(10, i));

digits[digit].Add(data[j]);

}

int index = 0;

for (int k = 0; k < digits.Count; k++)

{

IList<int> selDigit = digits[k];

for (int l = 0; l < selDigit.Count; l++)

{

data[index++] = selDigit[l];

}

}

ClearDigits();

}

printSortedData();

}

private void ClearDigits()

{

for (int k = 0; k < digits.Count; k++)

{

digits[k].Clear();

}

}

public void printSortedData()

{

Console.WriteLine("The Sorted Numbers are : ");

for (int i = 0; i < data.Length; i++)

{

Console.WriteLine(data[i]);

}

}

**Program Class:**

static void Main(string[] args)

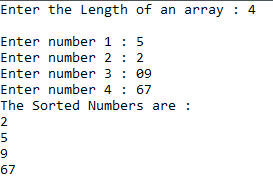
{

Radix obj= new Radix();

obj.RadixSort();

}

**Output:**



**Question 2:** **Write a program to create a static tree.**

**Code:**

static void Main(string[] args)

{

int[] array = new int[5];

for (int i = 0; i < array.Length; i++)

{

Console.WriteLine("Enter value to add");

int element = int.Parse(Console.ReadLine());

if (array[i] == 0)

{

array[i] = element;

continue;

}

else if (array[(2 \* i) + 1] == 0)

{

array[(i \* 2) + 1] = element;

continue;

}

else if (array[(2 \* i) + 2] == 0)

{

array[(i \* 2) + 2] = element;

continue;

}

}

Console.Write(array[0] + " ");

for (int i = 0; i < array.Length; i++)

{

if ((2 \* i) + 1 >= array.Length)

{

return;

}

else

{

Console.Write(array[(2 \* i) + 1] + " ");

}

if ((2 \* i) + 2 >= array.Length)

{

return;

}

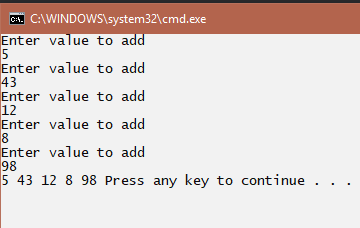
else

{

Console.Write(array[(2 \* i) + 2] + " ");

}

}

**Output:**