**Question 1: Implement the concept of B-Tree**

**Code:**

**Program Class:**

static void Main(string[] args)

{

Tree obj=new Tree();

Console.WriteLine("How many values you want to add");

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

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

{

Console.WriteLine("Enter value");

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

obj.AddRc(val);

}

string ans = "";

obj.Print(null, ref ans);

Console.WriteLine(ans);

}

**BTree Class:**

class Tree

{

Node top;

public Tree()

{

top = null;

}

public Tree(int val)

{

top = new Node(val);

}

public void Print(Node N,ref string temp)

{

if (N == null)

{

N = top;

}

if (N.left != null)

{

Print(N.left, ref temp);

temp = temp + N.data.ToString().PadLeft(3);

}

else

{

temp = temp + N.data.ToString().PadLeft(3);

}

if (N.right != null)

{

Print(N.right,ref temp);

}

}

public void AddRc(int val)

{

AddR(ref top, val);

}

public void Add(int val)

{

if (top == null)

{

Node n = new Node(val);

top = n;

return;

}

Node current = top;

bool Added = false;

do

{

if (val < current.data)

{

if (current.left == null)

{

Node n = new Node(val);

current.left = n;

Added = true;

continue;

}

else

{

current = current.left;

}

}

if (val >= current.data)

{

if (current.right == null)

{

Node n = new Node(val);

current.right = n;

Added = true;

continue;

}

else

{

current = current.right;

}

}

} while (!Added);

}

private void AddR(ref Node N, int val)

{

if (N == null)

{

Node newNode = new Node(val);

N = newNode;

return;

}

if (val < N.data)

{

AddR(ref N.left, val);

return;

}

if (val >= N.data)

{

AddR(ref N.right, val);

return;

}

}

}

**Node Class:**

public Node(int val)

{

data = val;

left = null;

right = null;

}

**Output:**

