Daily Practice Work:- Date: 22/8/2023

Program:- Find Largest Value In Each Tree

```
package com.ishwarchavan;
import java.util.*;

public class LargestValue{          //CLASS IS CREATED

static class Node{
      int val;
      Node left, right; // Two pointer is created
};

static void helper(Vector<Integer> res, Node root, int d) {             //Function is
created
      if (root == null)
            return;
      if (d == res.size())          //IF true then executed below statment
            res.add(root.val);
      else                                          //IF false then executed below statment
            res.set(d, Math.max(res.get(d), root.val));
      helper(res, root.left, d + 1);
      helper(res, root.right, d + 1);
}

static Vector<Integer> largestValues(Node root) {           //function to find largest
values
      Vector<Integer> res = new Vector<>();
      helper(res, root, 0);
      return res;             //return res
}
static Node newNode(int data) {
      Node temp = new Node();       //temp object is created
      temp.val = data;
      temp.left = temp.right = null;
      return temp;                  //return temp
}
public static void main(String[] args)    {           //MAIN progam is started
      Node root = null;
      root = newNode(10);
      root.left = newNode(5);
      root.right = newNode(2);
      root.left.left = newNode(9);
      root.left.right = newNode(2);
      root.right.right = newNode(7);

      Vector<Integer> res = largestValues(root);

      for (int i = 0; i < res.size(); i++)   //iterating loop
                  System.out.print(res.get(i)+" ");   //calling and printing value
    }
}
```

Program:- Sort Character By Frequency

```
package com.ishwarchavan;
import java.util.*;
import java.util.Collections;
import java.util.Map;

class SortCharacter {                     //class created

static int countFrequency(String str, char ch) {            //contFrequency function is
created
```

```
	int count = 0;                //count varialbe declare and initialize

	for(int i=0; i<str.length(); i++) {             //loop iterating
		if(str.charAt(i) == ch) {
			++count;                        /increment the count
		}
	}
	return count;           //return count
}
	static void sortArr(String str) {         //sortArr function is created for
sorting string in ascending order

		int n = str.length();
		Map<Character, Integer> freqDict= new HashMap<Character, Integer>();
	//dictionary is created for store the frequency of characters

		for(int i= 0; i<n; i++) {                              //loop iterating to
count the frequency
			if(freqDict.containsKey(str.charAt(i))) {
				freqDict.put(str.charAt(i),freqDict.get(str.charAt(i)));

			}
			else {                  //otherwise execute this statement
				freqDict.put(str.charAt(i), 1);

			}
		}
		List<Map.Entry<Character, Integer>> sortedDict= new
ArrayList<Map.Entry<Character, Integer>>(       //sore the dictionary in ascending
order
			freqDict.entrySet());
			Collections.sort(sortedDict, new Comparator<Map.Entry<Character,
Integer>>(){
				public int compare(Map.Entry<Character,Integer>
o1,Map.Entry<Character,Integer> o2) {
					return (o1.getValue()== (o2.getValue()))? o1.getKey()-
o2.getKey(): o1.getValue()- o2.getValue();            //return with condition operator
				}

			});
			for(Map.Entry<Character, Integer> entry : sortedDict) {
				for(int i = 0; i<entry.getValue(); i++) { //IF TRUE THEN print
below statement
					System.out.println(entry.getKey());
				}
			}
		}
		public static void main(String[]args){          //main program is started
			String str= "Aabb";
			sortArr(str);           //call the function
		}
	}
```