Niraj_Patil:

Q1)

a)

Code:

```
package com.sort.ds;

import java.util.Arrays;

public class QuickSort {


    public static void quick(int arr[],int
left,int right) {
        if(left>=right)
            return;

        int i=0;int j=right; int pivot;
        while(i<j) {
            pivot=arr[left];
        while(arr[i]<=pivot&&i<right)
        {
            i++;
        }
        while(arr[j]>pivot) {
            j--;
        }
        if(i<j) {
            swap(arr,i,j);
        }
        }
        swap(arr,j,left);
        quick(arr, left,j-1);

        quick(arr,j+1,right);

            }
```

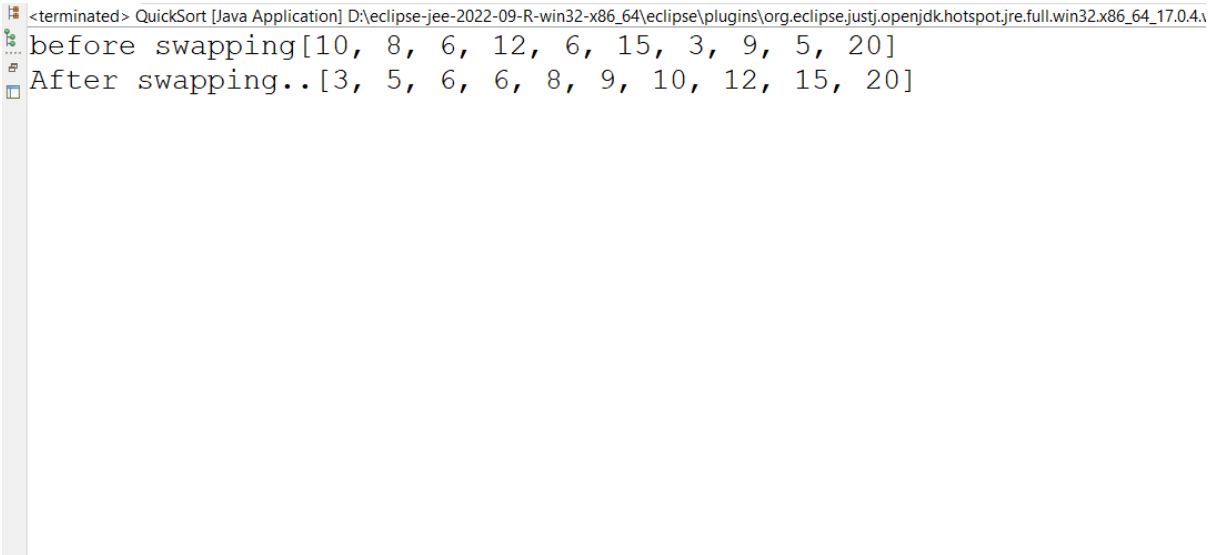
```
public static void swap(int arr[],int a,int b)
{
    int temp=arr[a];
    arr[a]=arr[b];
    arr[b]=temp;
}

    public static void main(String[] args) {
        QuickSort a=new QuickSort();
        int arr[]= {10,8,6,12,6,15,3,9,5,20};

        System.out.println("before
swapping"+Arrays.toString(arr));
        quick(arr,0,arr.length-1);
        System.out.println("After
swapping.."+Arrays.toString(arr));
    }

}
```

Output:



b)

code:

```
package com.exam.app;
```

```java
import java.util.Arrays;
import java.util.Scanner;



public class PreOrderTraversal {
    public static class Node {
        private int data;
        Node left;
        Node right;

        public Node() {
            data = 0;
            left = null;
            right = null;
        }

        public Node(int data) {
            this.data = data;
            left = null;
            right = null;
        }
    }// node class

    private Node root;

    public PreOrderTraversal() { // constructor
        root = null;
    }

    public void preorder(Node trav) {// base
logic for preorder
        if (trav == null)
            return;
        System.out.print(trav.data + " ");
        preorder(trav.left);
        preorder(trav.right);
```

```
    }

    public void add(int val) {
        Node newNode = new Node(val);
        if (root == null)
            root = newNode;
        else {
            Node trav = root;
    while (true) {
        if (val < trav.data)
        {
        if (trav.left != null)
        {
            trav = trav.left;

                    }
        else {
            trav.left = newNode;
                break;
            }
            } else {
                    if (val >= trav.data) {
                        if (trav.right != null)
{
                            trav = trav.right;
                        } else {
                            trav.right =
newNode;
                            break;
                        }
                    }
                }
            }

        }
    }
```

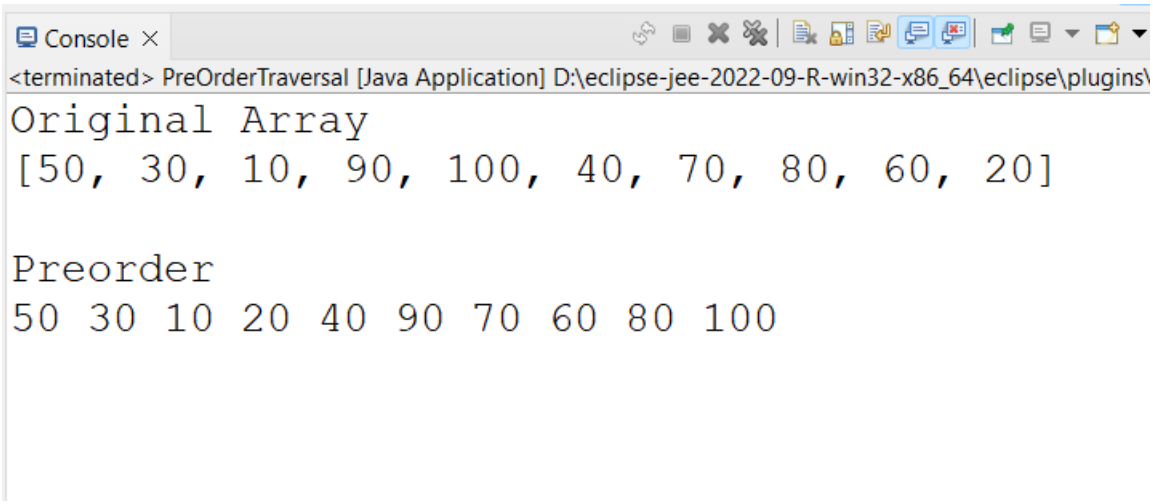
```
    public static void main(String[] args) {
        PreOrderTraversal bst = new
PreOrderTraversal();
        Scanner sc = new Scanner(System.in);
        bst.add(50);
        bst.add(30);
        bst.add(10);
        bst.add(90);
        bst.add(100);
        bst.add(40);
        bst.add(70);
        bst.add(80);
        bst.add(60);
        bst.add(20);
        int arr[] = { 50, 30, 10, 90, 100, 40,
70, 80, 60, 20 };// same array
        System.out.println("Original Array");

    System.out.println(Arrays.toString(arr));

        System.out.println();
        System.out.println("Preorder");
        bst.preorder(bst.root);
        System.out.println();
    }

}
```

output:


Console
<terminated> PreOrderTraversal [Java Application] D:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\
Original Array
[50, 30, 10, 90, 100, 40, 70, 80, 60, 20]
Preorder
50 30 10 20 40 90 70 60 80 100