

1. Write a Java program to

a. Perform quick sort

```
package Sorting;

import java.util.Arrays;

public class QuickSort1 {

    public void swap(int arr[],int x,int y) {

        int temp=arr[x];

        arr[x]=arr[y];

        arr[y]=temp;

    }

    public int partition(int[] arr,int start,int end) {

        int pivot=arr[start];

        int count=0;

        for(int i=start+1;i<=end;i++) {

            if(pivot>=arr[i]) {

                count++;

            }

        }

        int pvot_idx=start+count;

        swap(arr,start,pvot_idx);

        int i=start;

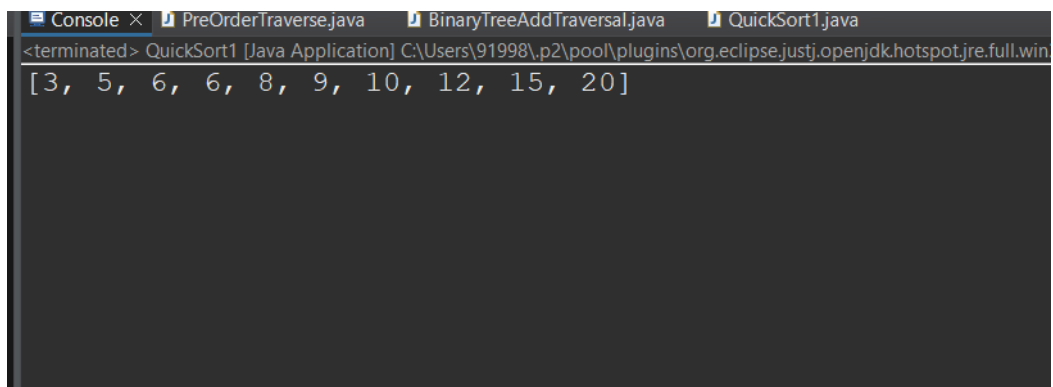
        int j=end;
```

```
while(i<pvot_idx && j>pvot_idx) {
    if(pivot>=arr[i]) {
        i++;
    }
    else if(pivot<=arr[j]) {
        j--;
    }
    else {
        swap(arr,i,j);
    }
}
return pvot_idx;
}

public void QuickSort(int arr[],int start,int end) {
    if(start<end) {
        int j=partition(arr, start, end);
        QuickSort(arr,start,j-1);
        QuickSort(arr, j+1, end);
    }
}

public static void main(String[] args) {
    int arr[] = { 10, 8, 6, 12, 6, 15, 3, 9, 5, 20 };
    QuickSort1 qs = new QuickSort1();
}
```

```
qs.QuickSort(arr, 0, arr.length-1);  
  
System.out.println(Arrays.toString(arr));  
  
}  
  
}
```



The screenshot shows the Eclipse IDE's console window. The title bar includes tabs for 'Console', 'PreOrderTraverse.java', 'BinaryTreeAddTraversal.java', and 'QuickSort1.java'. The console text shows the application has terminated and displays the sorted array: [3, 5, 6, 6, 8, 9, 10, 12, 15, 20].

```
<terminated> QuickSort1 [Java Application] C:\Users\91998\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win  
[3, 5, 6, 6, 8, 9, 10, 12, 15, 20]
```

b. Perform preorder tree traversal

```
package BinaryTree;

public class PreOrderTraverse {

    Node root;

    static public class Node{

        int key;

        Node left, right;

        public Node(int key) {

            this.key=key;

        }

        @Override

        public String toString() {

            return "Node [key=" + key + "]";

        }

    }

    public void preOrder(Node root) {

        if (root==null) {
```

```
return;

}

System.out.print(" "+root.key);

preOrder(root.left);

preOrder(root.right);

}

public static void main(String[] args) {

PreOrderTraverse pr=new PreOrderTraverse();

pr.root=new Node(6);

pr.root.left=new Node(4);

pr.root.left.left=new Node(3);

pr.root.left.right=new Node(5);

pr.root.right=new Node(8);

pr.root.right.left=new Node(7);

pr.root.right.right=new Node(9);

System.out.println("\nPreOrder");

pr.preOrder(pr.root);

}

}
```

```
....
Console x PreOrderTraverse.java BinaryTreeAddTrav
<terminated> PreOrderTraverse [Java Application] C:\Users\919988
PreOrder
6 4 3 5 8 7 9|
```