Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a program to search a given value in sorted array using binary search technique.

```
package ds2021;
import java.util.Scanner;
public class BinarySearch {
    int arr[];
    int size;
    int len;
    BinarySearch(int size){
        this.size=size;
        arr = new int[size];
    }
    BinarySearch(){
        this.size=10;
        arr = new int[10];
    }
    boolean isFull(){
        if(len>size-1)
            return true;
        else
            return false;
    }
    boolean isEmpty(){
        if(len<0)
            return true;
        else
            return false;
    }
    public void Insert(int val){
        int i;
        for(i=0;i<len &&arr[i]<=val ;i++){}
        len++;
        for(int j=len-1;j>i;j--){
            arr[j]= arr[j-1];
        }

        arr[i] = val;
    }
    public void BinSearch(int first, int last, int key){
        int mid = (first + last)/2;
        while( first <= last ){
            if ( arr[mid] < key ){
                first = mid + 1;
            }
            else if ( arr[mid] == key ){
                System.out.println("Element is found at index: " + mid);
                break;
            }
            else{
                last = mid - 1;
            }
            mid = (first + last)/2;
        }
        if ( first > last ){
            System.out.println("Element is not found!");
        }
    }
    void Display(){
        System.out.println("Array Elements : ");
        for(int i=0;i<len;i++){
            System.out.println(arr[i]);
        }
    }
    public static void main(String arg[]){
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter Size of Array : ");
        int size = scn.nextInt();
        BinarySearch bs = new BinarySearch(size);
        char c;
        int val;
        String s;
        while(true){
```

```
System.out.println("\nOptions");
System.out.println("================");
System.out.println("1 - Insert");
System.out.println("2 - Binary Search");
System.out.println("3/<p> - Display");
System.out.println("0/<q> - Exit");
System.out.println("Enter Your Choice : ");
s = scn.next();
c = s.charAt(0);
switch(c){
    case '1':
        if(bs.isFull()){
            System.out.println("Array is Full....");
            break;
        }
        System.out.println("ENter value : ");
        val = scn.nextInt();
        bs.Insert(val);
        break;
    case '2':
        if(bs.isEmpty()){
            System.out.println("Array is Empty....");
            break;
        }
        System.out.println("ENter Key value : ");
        val = scn.nextInt();
        bs.BinSearch(0,bs.len,val);
        break;
    case 'p':
    case '3':
        if(bs.isEmpty()){
            System.out.println("Array is Empty....");
            break;
        }
        bs.Display();
```

```
                    break;
                case 'q':
                case '0':
                    System.exit(0);
                    break;
                default:
                    System.out.println("Please enter valid choice....");
                    break;
            }
        }
    }
}
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program to perform following operations on Binary Tree.

1) **Create a binary tree**
2) **Search and replace a given key in binary tree.**
3) **Traverse a tree in preorder**
4) **Traverse a tree in inorder**
5) **Traverse a tree in postorder**

```
package ds2021;

import java.util.Scanner;

public class Tree {

    class Node {

        int value;

        Node left;

        Node right;


        Node(int value) {

            this.value = value;

            right = null;

            left = null;

        }

    }

    Node root = null;


    Scanner scn = new Scanner(System.in);

    Node CreateTree(){

        System.out.println("Enter Key value : ");

        int val = scn.nextInt();

        if(val == 0){

            return null;

        }

        Node nn;

        nn = new Node(val);

        System.out.println("Enter Left Child of "+val+" Or Enter 0 : ");

        nn.left = CreateTree();

        System.out.println("Enter Right Child of "+val+" Or Enter 0 : ");

        nn.right = CreateTree();

        return nn;

    }
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program to perform following operations on Binary Tree.

1) **Create a binary tree**
2) **Search and replace a given key in binary tree.**
3) **Traverse a tree in preorder**
4) **Traverse a tree in inorder**
5) **Traverse a tree in postorder**

```
public void Replace(Node root,int key,int newval){
    root.value = newval;
        System.out.println("\nValue Replaced..");
}
public void Search(Node root,int key,int newval){
    if(root != null){
        if(root.value == key){
            System.out.println("\nValue Searched...");
            Replace(root,key,newval);
            return;
        }
        Search(root.left,key,newval);
        Search(root.right,key,newval);
    }
}
public void Inorder(Node root){
    if(root != null){
        Inorder(root.left);
        System.out.print(root.value + "  ");
        Inorder(root.right);


    }
}
public void Preorder(Node root){
    if(root != null){
        System.out.print(root.value + "  ");
        Preorder(root.left);
        Preorder(root.right);
    }
}
public void Postorder(Node root){
    if(root != null){
        Postorder(root.left);
        Postorder(root.right);
        System.out.print(root.value + "  ");

    }
}

public static void main(String arg[]){
    Tree bt = new Tree();
    int ch = 0;
    while(ch != 6){
        System.out.println("\n====================");
        System.out.println("Option");
        System.out.println("====================");
        System.out.println("1. Create Tree");
        System.out.println("2. Search and Replace");
        System.out.println("3. View Inorder");
        System.out.println("4. View Preorder");
        System.out.println("5. View Postorder");
        System.out.println("6. Exit");
        System.out.println("====================");
        System.out.println("Enter Your Choice:");
        Scanner sc = new Scanner(System.in);
        ch = sc.nextInt();
        switch(ch){
            case 1:
                bt.root = bt.CreateTree();
                break;
            case 2:
                System.out.println("Enter Search Key : ");
                int key = sc.nextInt();
                System.out.println("Enter New Value : ");
                int newval = sc.nextInt();
                bt.Search(bt.root,key,newval);



                break;
            case 3:
                if(bt.root == null){
                    System.out.println("\nEmpty.....");
                }
                else{
                    System.out.println("\n=====================");
                    System.out.println("Inorder : ");
                    bt.Inorder(bt.root);
                }
                break;
            case 4:
                if(bt.root == null){
                    System.out.println("\nEmpty.....");
                }
                else{
                    System.out.println("\n=====================");
                    System.out.println("Preorder : ");
                    bt.Preorder(bt.root);
                }
                break;
            case 5:
                if(bt.root == null){
                    System.out.println("\nEmpty.....");
                }
```

```
                    else{
                        System.out.println("\n====================");
                        System.out.println("Postorder : ");
                        bt.Postorder(bt.root);
                    }
                    break;
                case 6:
                    System.exit(0);
                    break;
                default :
                    System.out.println("\nEnter Valid Choice...");
                    break;
            }
        }
    }
}
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. Write a menu driven program to perform following operations on Binary Search Tree.

1) **Insert a key**
2) **Display all keys in ascending order (inorder traversal).**
3) **Display all keys in descending order. (Converse inorder traversal).**

```
package ds2021;
import java.util.Scanner;
public class BST {
    class Node {
        int value;
        Node left;
        Node right;

        Node(int value) {
            this.value = value;
            right = null;
            left = null;
        }
    }
    Node root = null;

    public void add(int value) {
        root = addRecursive(root, value);
    }
    private Node addRecursive(Node current, int value) {
        if (current == null) {
            return new Node(value);
        }

        if (value < current.value) {
            current.left = addRecursive(current.left, value);
        }
        else if (value > current.value) {
            current.right = addRecursive(current.right, value);
        }
        else {
            return current;
        }


        return current;
    }


    public void Inorder(Node root){
        if(root != null){
            Inorder(root.left);
            System.out.print(root.value + "  ");
            Inorder(root.right);


        }
    }
    public void ConInorder(Node root){
        if(root != null){
            ConInorder(root.right);
            System.out.print(root.value + "  ");
            ConInorder(root.left);


        }
    }


    public static void main(String arg[]){
        BST bt = new BST();
        int ch = 0;
        while(ch != 4){
            System.out.println("\n====================");
            System.out.println("Options");
            System.out.println("====================");
            System.out.println("1. Insert");
            System.out.println("2. View Ascending (Inorder)");
            System.out.println("3. View Descending (Converse Inorder)");
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. Write a menu driven program to perform following operations on Binary Search Tree.

1) **Insert a key**
2) **Display all keys in ascending order (inorder traversal).**
3) **Display all keys in descending order. (Converse inorder traversal).**

```
System.out.println("4. Exit");
System.out.println("====================");
System.out.println("Enter Your Choice:");
Scanner sc = new Scanner(System.in);
ch = sc.nextInt();
switch(ch){
    case 1:
        System.out.println("\nEnter Key : ");
        int data = sc.nextInt();
        bt.add(data);

        break;
    case 2:
        if(bt.root == null){
            System.out.println("\nEmpty.....");
        }
        else{
            System.out.println("\n====================");
            System.out.println("Ascending (Inorder) : ");
            bt.Inorder(bt.root);
        }
        break;
    case 3:
        if(bt.root == null){
            System.out.println("\nEmpty.....");
        }
        else{
            System.out.println("\n====================");
            System.out.println("Descending (Converse Inorder) : ");
            bt.ConInorder(bt.root);
        }
        break;
        case 4:
            System.exit(0);
            break;
        default :
            System.out.println("\nEnter Valid Choice...");
            break;
    }
}
```