

B1

```
import  
java.util.ArrayList;  
  
import java.util.Collections;  
import java.util.Scanner;  
  
public class Array_value{  
    Scanner sc = new Scanner(System.in);  
  
    private ArrayList<Integer> arrayList;  
  
    public Array_value() {  
        this.arrayList = new ArrayList<Integer>();  
    }  
  
    public void GetValue(){  
        System.out.print("Nhập Số lượng phần tử: ");  
        int n = sc.nextInt();  
        for (int i = 0; i < n; i++){  
            this.arrayList.add(sc.nextInt());  
        }  
    }  
}
```

```
public void SortUp(){

    Collections.sort(this.arrayList);

}

public void SortDown(){

    Collections.sort(this.arrayList);

    Collections.reverse(this.arrayList);

}

public void PrintArray(){

    for (int i = 0; i < this.arrayList.size() ; i++) {

        System.out.print(this.arrayList.get(i).toString()+" ");

    }

}

public static void main(String[] args) {

    Array_value arr = new Array_value();

    arr.GetValue();

    System.out.println("Xap xep tang");

    arr.SortUp();
```

```
        arr.PrintArray();

        System.out.println("\nXap xep Giam");
        arr.SortDown();
        arr.PrintArray();
    }
}
```

B2

```
public class CayTimKiemNhiPhan {
    public int Key;
    public CayTimKiemNhiPhan left,right;

    public CayTimKiemNhiPhan(int key){
        this.Key = key;
        this.left = null;
        this.right = null;
    }

    public static CayTimKiemNhiPhan
    addNode(CayTimKiemNhiPhan root,int node){
        if (root == null) {
```

```
    root = new CayTimKiemNhiPhan(node);

    return root;

}
```

```
if (root.Key > node){

    root.left = addNode(root.left,node);

}else if (root.Key < node){

    root.right = addNode(root.right,node);

}

return root;

}
```

//Duyệt trung thứ tự (In-order Traversal)

```
public void InOrDerTraversal
(CayTimKiemNhiPhan Node){

    if (Node != null){

        InOrDerTraversal(Node.left);

        System.out.print(Node.Key+" ");

        InOrDerTraversal(Node.right);

    }

}
```

// Duyệt tiên thứ tự (Pre-order Traversal)

```
    public void PreOrDerTraversal  
(CayTimKiemNhiPhan Node){  
  
        if (Node != null){  
  
            System.out.print(Node.Key+" ");  
  
            PreOrDerTraversal(Node.left);  
  
            PreOrDerTraversal(Node.right);  
  
        }  
  
    }  
  
}
```

// Duyệt hậu thứ tự (Post-order Traversal)

```
    public void PostOrDerTraversal  
(CayTimKiemNhiPhan Node){  
  
        if (Node != null){  
  
            PostOrDerTraversal(Node.left);  
  
            PostOrDerTraversal(Node.right);  
  
            System.out.print(Node.Key+" ");  
  
        }  
  
    }  
  
}
```

```
public static void main(String[] args) {
```

```
    CayTimKiemNhiPhan Node = new  
    CayTimKiemNhiPhan(25);  
  
    Node = addNode(Node, 15);
```

```
Node = addNode(Node, 50);
Node = addNode(Node, 10);
Node = addNode(Node, 22);
Node = addNode(Node, 35);
Node = addNode(Node, 70);
Node = addNode(Node, 4);
Node = addNode(Node, 12);
Node = addNode(Node, 18);
Node = addNode(Node, 24);
Node = addNode(Node, 31);
Node = addNode(Node, 44);
Node = addNode(Node, 66);
Node = addNode(Node, 90);
```

```
System.out.print("\nDuyệt tiền thứ tự: ");
Node.PreOrDerTraversal(Node);
System.out.print("\nDuyệt Trung thứ tự: ");
Node.InOrDerTraversal(Node);
System.out.print("\nDuyệt hậu thứ tự: ");
Node.PostOrDerTraversal(Node);
}
```

}

