

```

package assignment3;

import java.util.Scanner;

public class RearrangeArrayElements {

    public static void main(String[] args) {
        RearrangeArrayElements rearrange = new RearrangeArrayElements();
        rearrange.inputAcceptor();
    }

    public void inputAcceptor() {
        Scanner scan = new Scanner(System.in);
        int size= scan.nextInt();
        if(!inputArraySizeValidator(size))
        {
            displayResult(null);
            return;
        }
        int[] inputArray= new int[size];
        for(int i=0;i<size; i++)
        {
            inputArray[i]=scan.nextInt();
        }
        if(!inputArrayValidator(inputArray))
        {
            displayResult(null);
            return;
        }
        int[] resultArray = computeRearrangedArray(inputArray);
        displayResult(resultArray);
    }

    public boolean inputArraySizeValidator(int size)
    {
        return size > 0;
    }

```

```

}

public boolean inputArrayValidator(int[] input) {
    for (int i=1;i<input.length;i++)
    {
        if (input[i]<input[i - 1])
        {
            return false;
        }
    }
    return true;
}

public int[] computeRearrangedArray(int[] inputArray) {
    int[] resultArray = new int[inputArray.length];
    int first = 0;
    int last = inputArray.length - 1;
    boolean flag = true;
    for (int i = 0; i < inputArray.length; i++) {
        if (flag) {
            resultArray[i] = inputArray[last--];
        } else {
            resultArray[i] = inputArray[first++];
        }
        flag = !flag;
    }
    return resultArray;
}

public void displayResult(int[] outputArray) {
    if (outputArray==null) {
        System.out.println("Give proper input");
    } else {
        for (int i:outputArray) {
            System.out.print(i + " ");
        }
    }
}

```

```

    }

    System.out.println();

}

}

}

```

The screenshot shows the Eclipse IDE interface. The left pane displays the source code for `RearrangeArrayElements.java`. The code defines a package `assignment3`, imports `java.util.Scanner`, and contains a class `RearrangeArrayElements` with several methods: `main`, `inputAcceptor`, `inputArraySizeValidator`, and `inputArrayValidator`. The `main` method creates an instance of the class and calls `inputAcceptor`. The `inputAcceptor` method uses a `Scanner` to read input, validates the array size, and then validates the array elements. The `inputArraySizeValidator` method checks if the size is greater than 0. The `inputArrayValidator` method checks if the array is sorted in ascending order. The right pane shows the console output, which displays the input array `7 1 2 3 4 5 6 7` and the rearranged array `7 1 6 2 5 3 4`.

```

1 package assignment3;
2 import java.util.Scanner;
3
4 public class RearrangeArrayElements {
5     public static void main(String[] args) {
6         RearrangeArrayElements rearrange = new RearrangeArrayElements();
7         rearrange.inputAcceptor();
8     }
9     public void inputAcceptor() {
10        Scanner scan = new Scanner(System.in);
11        int size= scan.nextInt();
12        if(!inputArraySizeValidator(size))
13        {
14            displayResult(null);
15            return;
16        }
17        int[] inputArray= new int[size];
18        for(int i=0;i<size; i++)
19        {
20            inputArray[i]=scan.nextInt();
21        }
22        if(!inputArrayValidator(inputArray))
23        {
24            displayResult(null);
25            return;
26        }
27        int[] resultArray = computeRearrangedArray(inputArray);
28        displayResult(resultArray);
29    }
30    public boolean inputArraySizeValidator(int size)
31    {
32        return size > 0;
33    }
34    public boolean inputArrayValidator(int[] input) {
35        for (int i=1;i<input.length;i++)
36        {
37            if (input[i]<input[i - 1])
38            {
39                return false;
40            }
41        }
42        return true;
43    }
44 }

```

```

<terminated> RearrangeArrayElements [Java Application] C:\Users\vineet.p2\pool\plugins\org.eclipse.justi.openjdk hotspot.jre.full.win32.x86
7 1 2 3 4 5 6 7
7 1 6 2 5 3 4

```

eclipse-workspace2 - Assignments/src/assignment3/RearrangeArrayElements.java - Eclipse IDE

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
```

RearrangeArrayElements.java X

```
1 package assignment3;
2 import java.util.Scanner;
3
4 public class RearrangeArrayElements {
5     public static void main(String[] args) {
6         RearrangeArrayElements rearrange = new RearrangeArrayElements();
7         rearrange.inputAcceptor();
8     }
9     public void inputAcceptor() {
10        Scanner scan = new Scanner(System.in);
11        int size= scan.nextInt();
12        if(inputArraySizeValidator(size))
13        {
14            displayResult(null);
15            return;
16        }
17        int[] inputArray= new int[size];
18        for(int i=0;i<size; i++)
19        {
20            inputArray[i]=scan.nextInt();
21        }
22        if(inputArrayValidator(inputArray))
23        {
24            displayResult(null);
25            return;
26        }
27        int[] resultArray = computeRearrangedArray(inputArray);
28        displayResult(resultArray);
29    }
30    public boolean inputArraySizeValidator(int size)
31    {
32        return size > 0;
33    }
34    public boolean inputArrayValidator(int[] input) {
35        for (int i=1;i<input.length;i++)
36        {
37            if (input[i]<input[i - 1])
38            {
39                return false;
40            }
41        }
42        return true;
43    }
```

Console X

```
<terminated> RearrangeArrayElements [Java Application] C:\Users\vinee\p2\pool\plugins\org.eclipse.justi.openjdkhotspotjre.full.win32.x86_64
0
Give proper input
```

eclipse-workspace2 - Assignments/src/assignment3/RearrangeArrayElements.java - Eclipse IDE

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
```

RearrangeArrayElements.java X

```
1 package assignment3;
2 import java.util.Scanner;
3
4 public class RearrangeArrayElements {
5     public static void main(String[] args) {
6         RearrangeArrayElements rearrange = new RearrangeArrayElements();
7         rearrange.inputAcceptor();
8     }
9     public void inputAcceptor() {
10        Scanner scan = new Scanner(System.in);
11        int size= scan.nextInt();
12        if(inputArraySizeValidator(size))
13        {
14            displayResult(null);
15            return;
16        }
17        int[] inputArray= new int[size];
18        for(int i=0;i<size; i++)
19        {
20            inputArray[i]=scan.nextInt();
21        }
22        if(inputArrayValidator(inputArray))
23        {
24            displayResult(null);
25            return;
26        }
27        int[] resultArray = computeRearrangedArray(inputArray);
28        displayResult(resultArray);
29    }
30    public boolean inputArraySizeValidator(int size)
31    {
32        return size > 0;
33    }
34    public boolean inputArrayValidator(int[] input) {
35        for (int i=1;i<input.length;i++)
36        {
37            if (input[i]<input[i - 1])
38            {
39                return false;
40            }
41        }
42        return true;
43    }
```

Console X

```
<terminated> RearrangeArrayElements [Java Application] C:\Users\vinee\p2\pool\plugins\org.eclipse.justi.openjdkhotspotjre.full.win32.x86_64
1 4
4
```