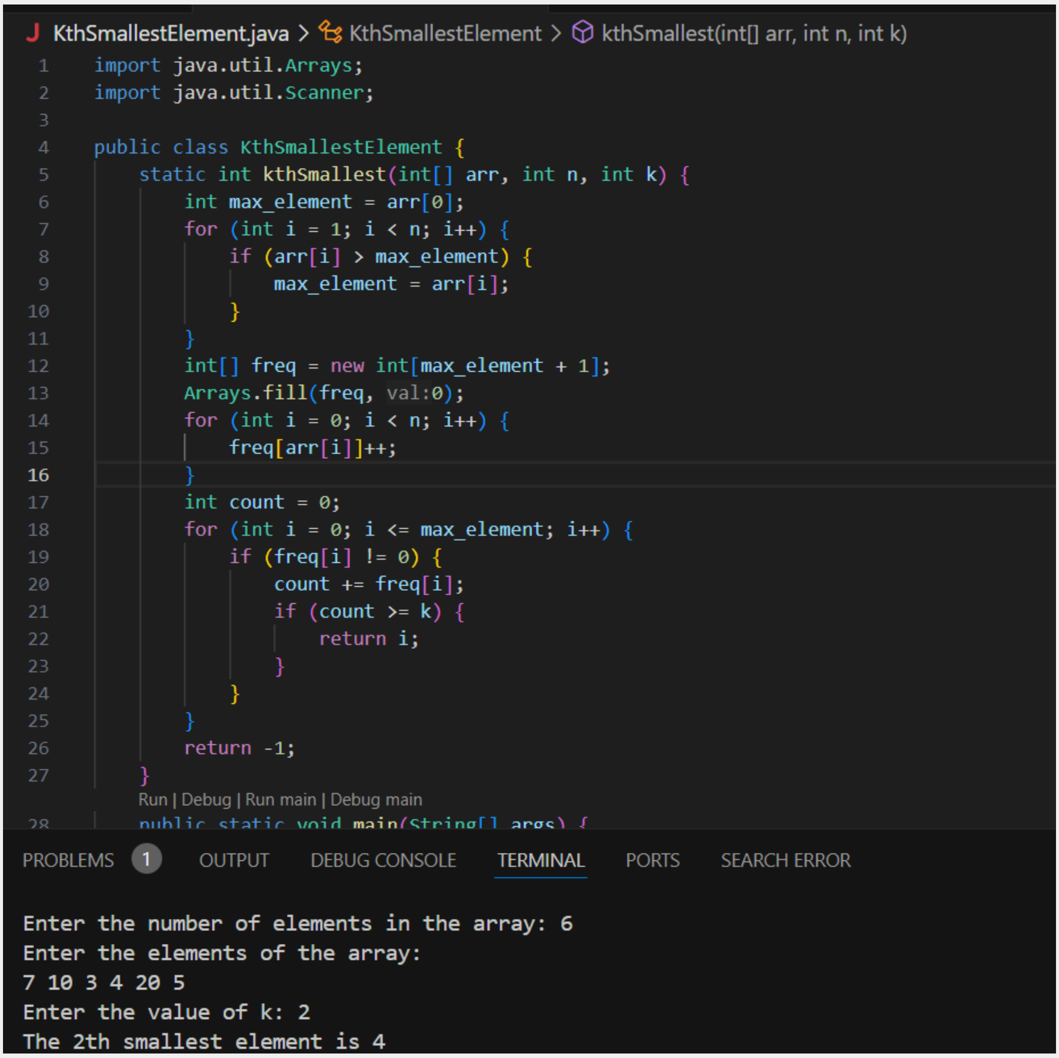
Kth Smallest Element

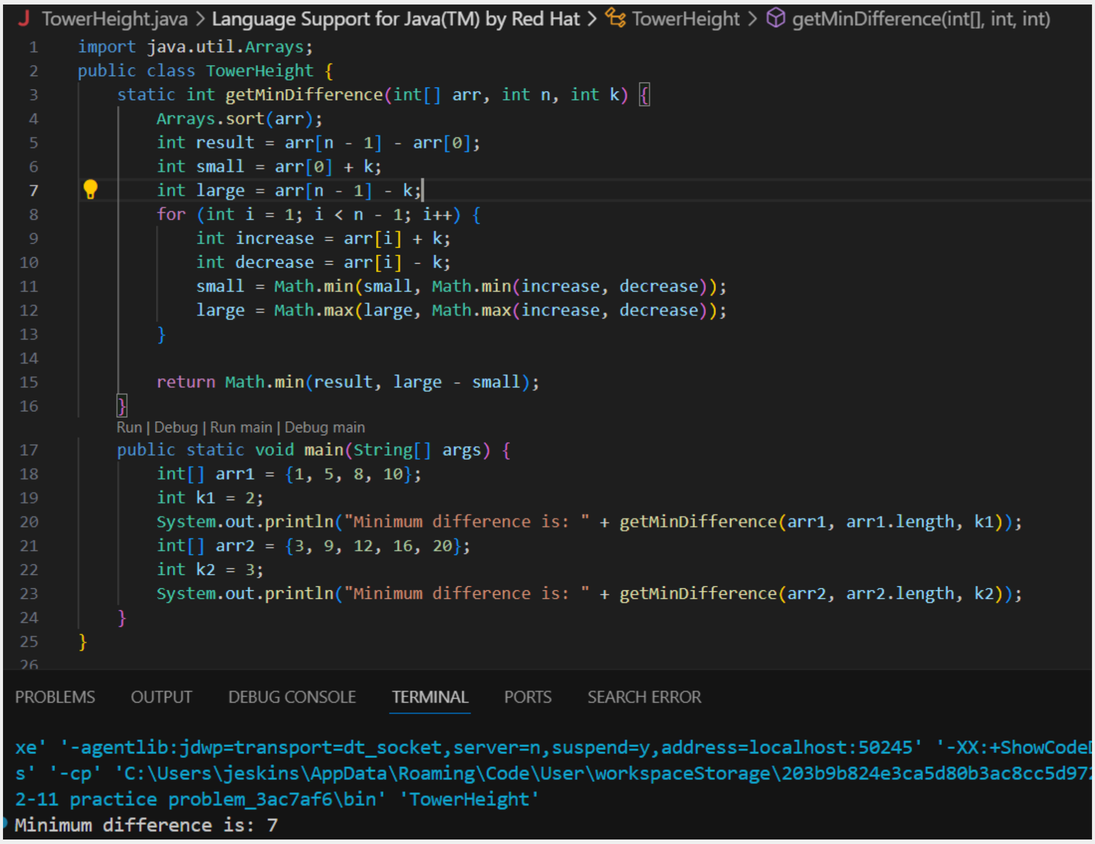


KthSmallestElement.java > KthSmallestElement > kthSmallest(int[] arr, int n, int k)
Run | Debug | Run main | Debug main
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH ERROR

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

public class KthSmallestElement {
    static int kthSmallest(int[] arr, int n, int k) {
        int max_element = arr[0];
        for (int i = 1; i < n; i++) {
            if (arr[i] > max_element) {
                max_element = arr[i];
            }
        }
        int[] freq = new int[max_element + 1];
        Arrays.fill(freq, val:0);
        for (int i = 0; i < n; i++) {
            freq[arr[i]]++;
        }
        int count = 0;
        for (int i = 0; i <= max_element; i++) {
            if (freq[i] != 0) {
                count += freq[i];
                if (count >= k) {
                    return i;
                }
            }
        }
        return -1;
    }
    public static void main(String[] args) {
```

```
Enter the number of elements in the array: 6
Enter the elements of the array:
7 10 3 4 20 5
Enter the value of k: 2
The 2th smallest element is 4
```

Minimize the height II

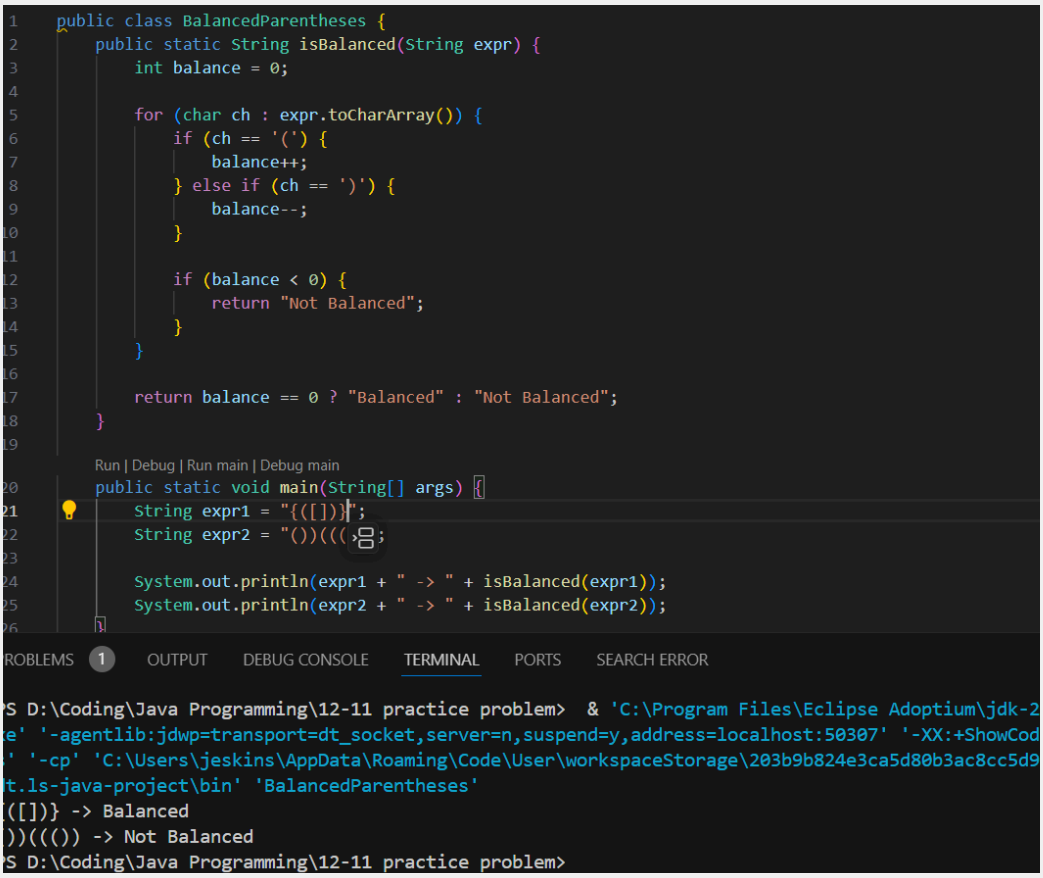


TowerHeight.java > Language Support for Java(TM) by Red Hat > TowerHeight > getMinDifference(int[], int, int)
Run | Debug | Run main | Debug main
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH ERROR

```java
import java.util.Arrays;
public class TowerHeight {
    static int getMinDifference(int[] arr, int n, int k) {
        Arrays.sort(arr);
        int result = arr[n - 1] - arr[0];
        int small = arr[0] + k;
        int large = arr[n - 1] - k;
        for (int i = 1; i < n - 1; i++) {
            int increase = arr[i] + k;
            int decrease = arr[i] - k;
            small = Math.min(small, Math.min(increase, decrease));
            large = Math.max(large, Math.max(increase, decrease));
        }

        return Math.min(result, large - small);
    }
    public static void main(String[] args) {
        int[] arr1 = {1, 5, 8, 10};
        int k1 = 2;
        System.out.println("Minimum difference is: " + getMinDifference(arr1, arr1.length, k1));
        int[] arr2 = {3, 9, 12, 16, 20};
        int k2 = 3;
        System.out.println("Minimum difference is: " + getMinDifference(arr2, arr2.length, k2));
    }
}
```

```
xe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:50245' '-XX:+ShowCode
s' '-cp' 'C:\Users\jeskins\AppData\Roaming\Code\User\workspaceStorage\203b9b824e3ca5d80b3ac8cc5d97
2-11 practice problem_3ac7af6\bin' 'TowerHeight'
Minimum difference is: 7
```

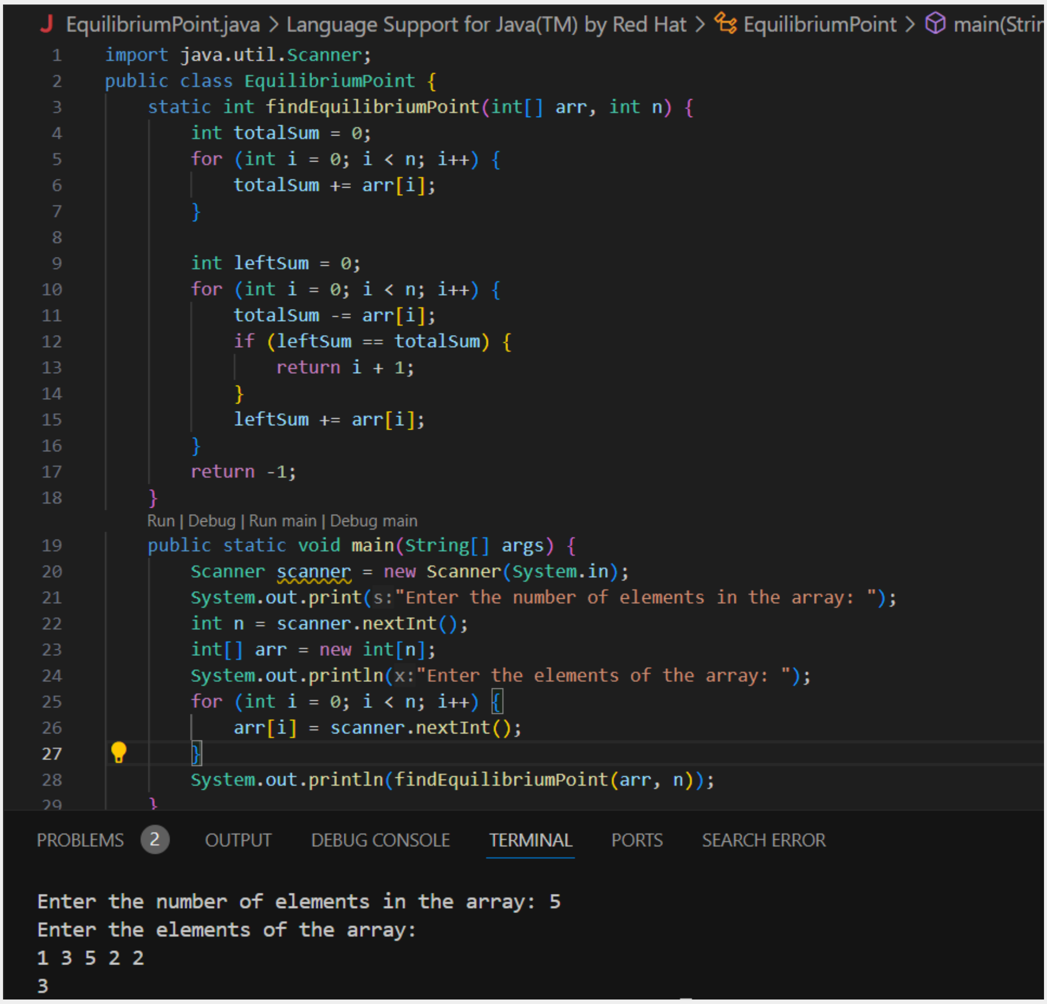
Parenthesis Checker



```java
public class BalancedParentheses {
    public static String isBalanced(String expr) {
        int balance = 0;

        for (char ch : expr.toCharArray()) {
            if (ch == '(') {
                balance++;
            } else if (ch == ')') {
                balance--;
            }

            if (balance < 0) {
                return "Not Balanced";
            }
        }

        return balance == 0 ? "Balanced" : "Not Balanced";
    }

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        String expr1 = "{([])}";
        String expr2 = "())(((   ;

        System.out.println(expr1 + " -> " + isBalanced(expr1));
        System.out.println(expr2 + " -> " + isBalanced(expr2));
    }
```

```
PS D:\Coding\Java Programming\12-11 practice problem>  & 'C:\Program Files\Eclipse Adoptium\jdk-2
e' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:50307' '-XX:+ShowCod
' '-cp' 'C:\Users\jeskins\AppData\Roaming\Code\User\workspaceStorage\203b9b824e3ca5d80b3ac8cc5d9
lt.ls-java-project\bin' 'BalancedParentheses'
{([])} -> Balanced
())((()) -> Not Balanced
PS D:\Coding\Java Programming\12-11 practice problem>
```

Equilibrium Point

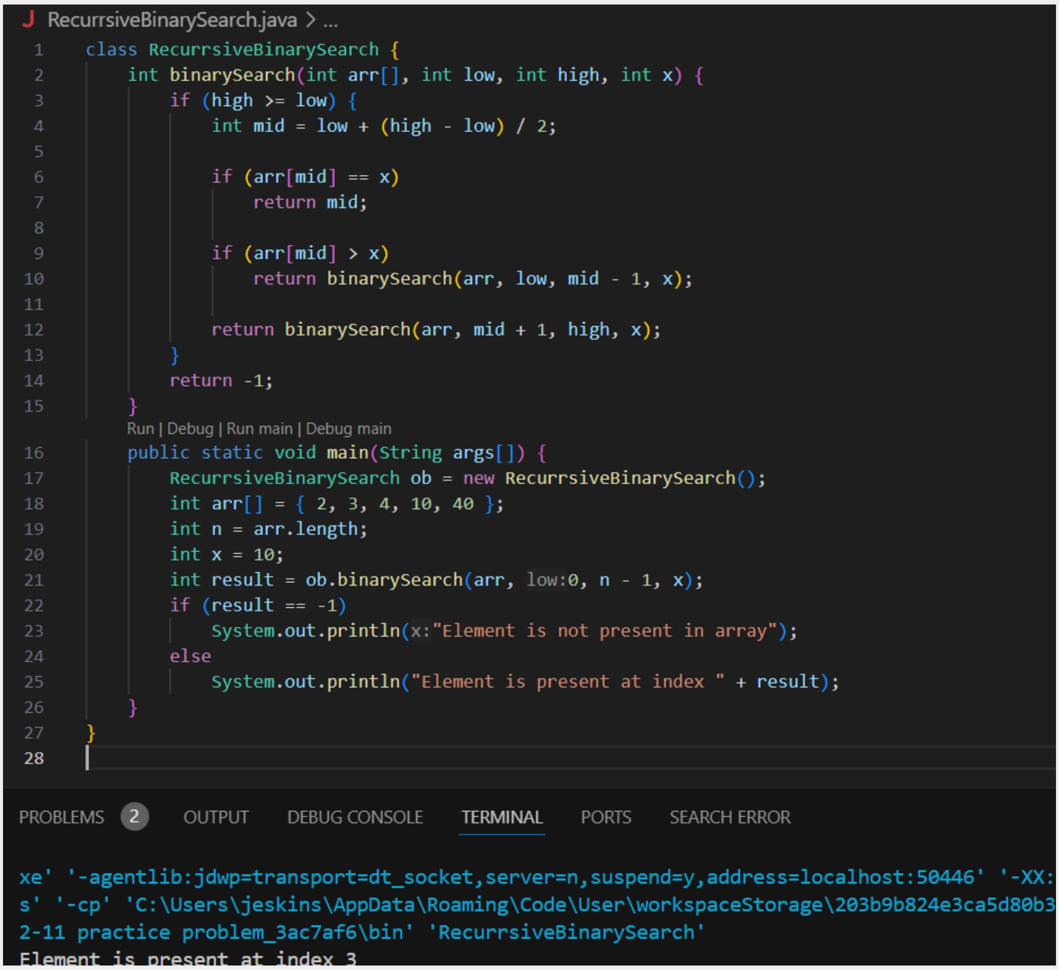


EquilibriumPoint.java > Language Support for Java(TM) by Red Hat > EquilibriumPoint > main(Strin

```java
import java.util.Scanner;
public class EquilibriumPoint {
    static int findEquilibriumPoint(int[] arr, int n) {
        int totalSum = 0;
        for (int i = 0; i < n; i++) {
            totalSum += arr[i];
        }

        int leftSum = 0;
        for (int i = 0; i < n; i++) {
            totalSum -= arr[i];
            if (leftSum == totalSum) {
                return i + 1;
            }
            leftSum += arr[i];
        }
        return -1;
    }
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println(x:"Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        System.out.println(findEquilibriumPoint(arr, n));
    }
```

```
Enter the number of elements in the array: 5
Enter the elements of the array:
1 3 5 2 2
3
```

Binary Search

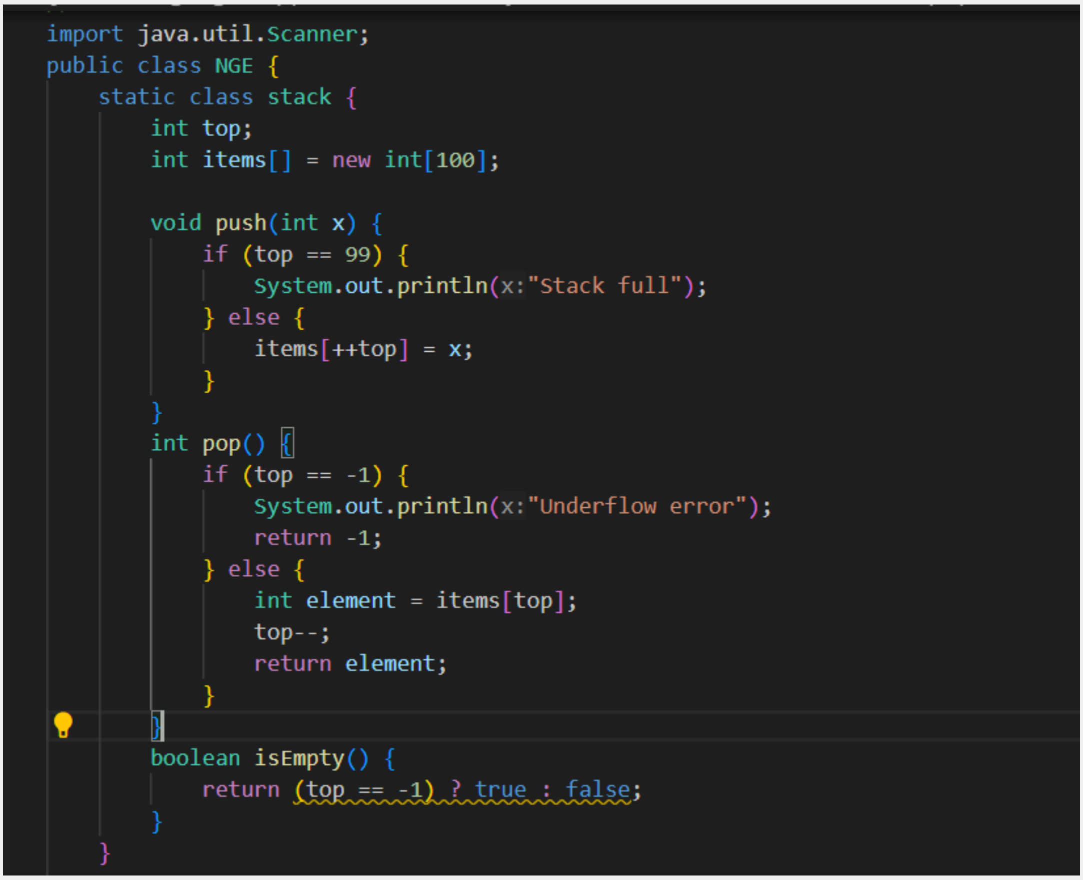


RecurrsiveBinarySearch.java > ...

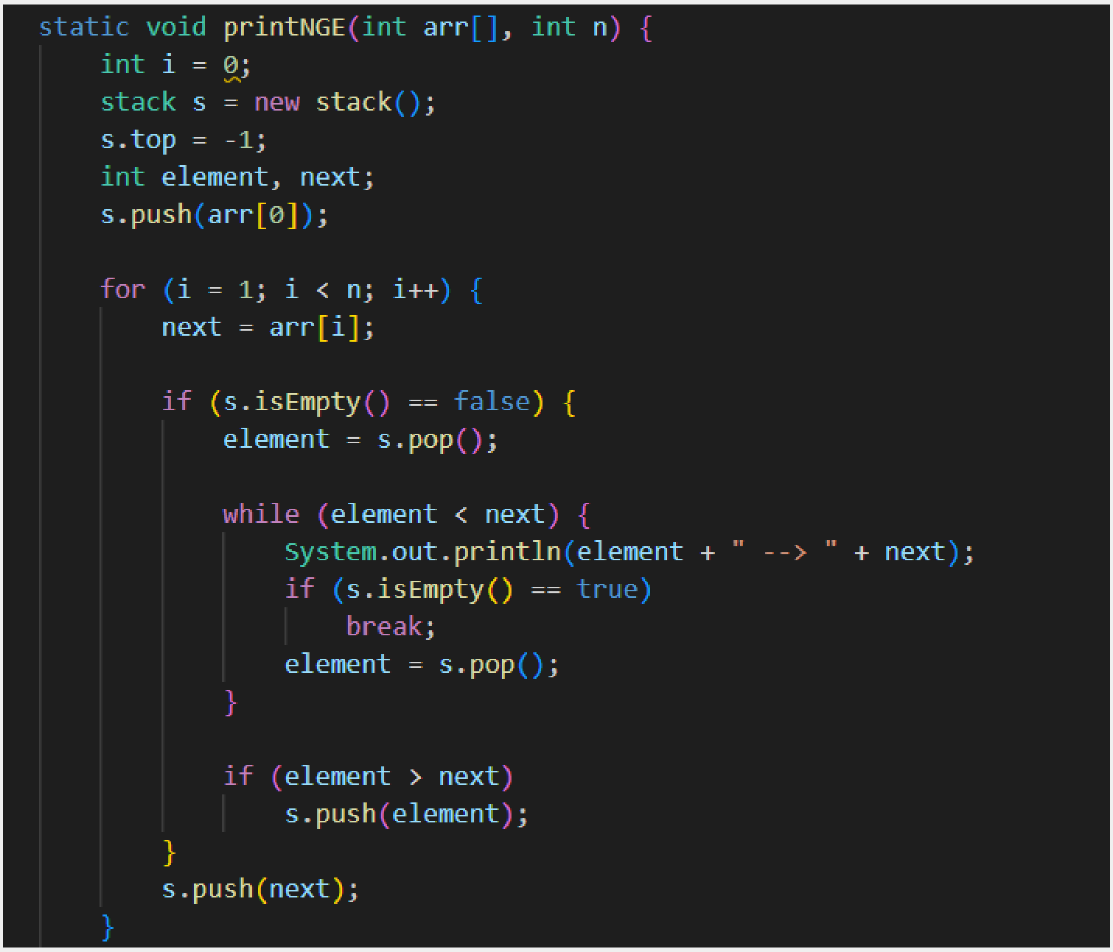
```java
class RecurrsiveBinarySearch {
    int binarySearch(int arr[], int low, int high, int x) {
        if (high >= low) {
            int mid = low + (high - low) / 2;

            if (arr[mid] == x)
                return mid;

            if (arr[mid] > x)
                return binarySearch(arr, low, mid - 1, x);

            return binarySearch(arr, mid + 1, high, x);
        }
        return -1;
    }
    Run | Debug | Run main | Debug main
    public static void main(String args[]) {
        RecurrsiveBinarySearch ob = new RecurrsiveBinarySearch();
        int arr[] = { 2, 3, 4, 10, 40 };
        int n = arr.length;
        int x = 10;
        int result = ob.binarySearch(arr, low:0, n - 1, x);
        if (result == -1)
            System.out.println(x:"Element is not present in array");
        else
            System.out.println("Element is present at index " + result);
    }
}

```

```
xe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:50446' '-XX:
s' '-cp' 'C:\Users\jeskins\AppData\Roaming\Code\User\workspaceStorage\203b9b824e3ca5d80b3
2-11 practice problem_3ac7af6\bin' 'RecurrsiveBinarySearch'
Element is present at index 3
```

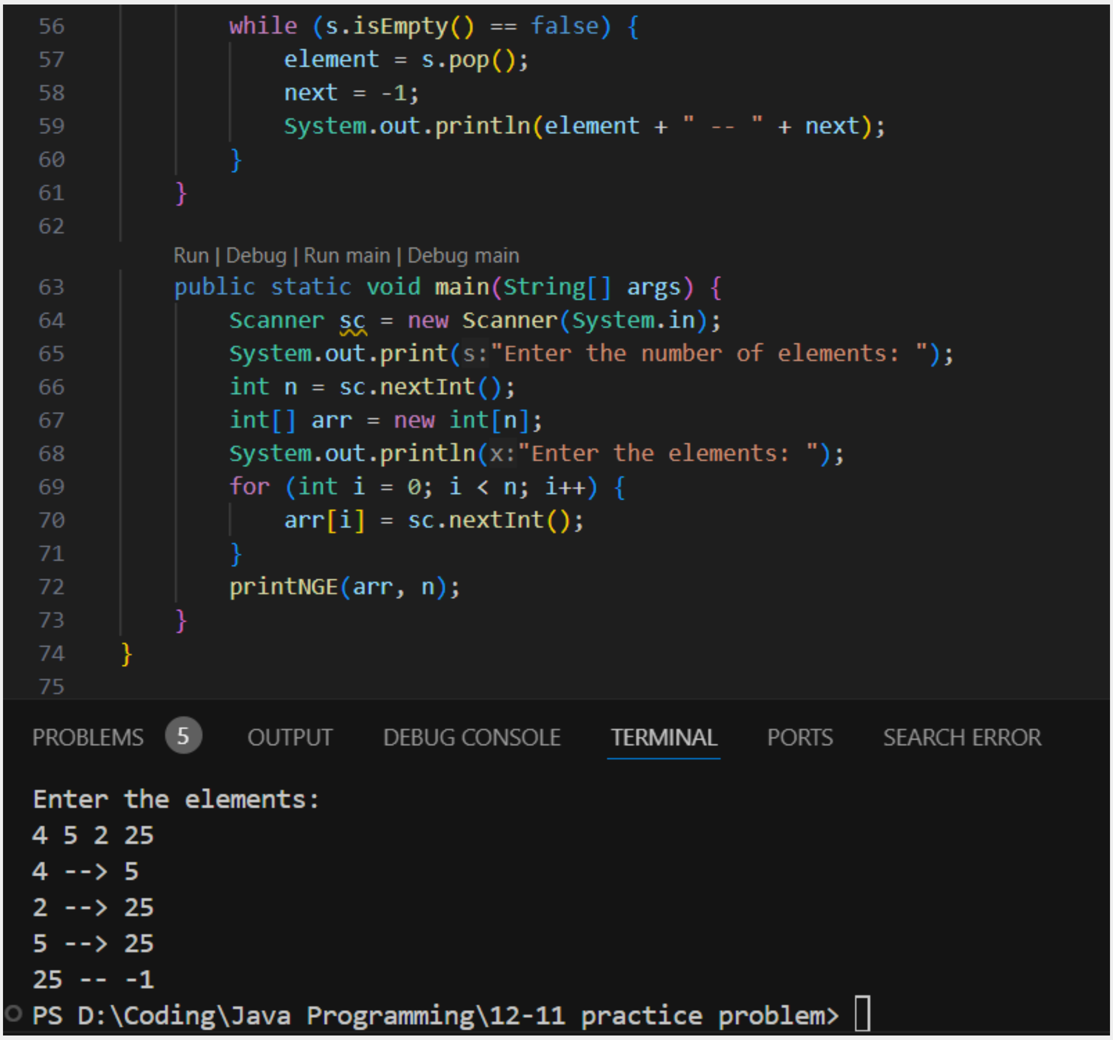
Next Greater Element



```java
import java.util.Scanner;
public class NGE {
    static class stack {
        int top;
        int items[] = new int[100];

        void push(int x) {
            if (top == 99) {
                System.out.println(x:"Stack full");
            } else {
                items[++top] = x;
            }
        }
        int pop() {
            if (top == -1) {
                System.out.println(x:"Underflow error");
                return -1;
            } else {
                int element = items[top];
                top--;
                return element;
            }
        }
        boolean isEmpty() {
            return (top == -1) ? true : false;
        }
    }
```



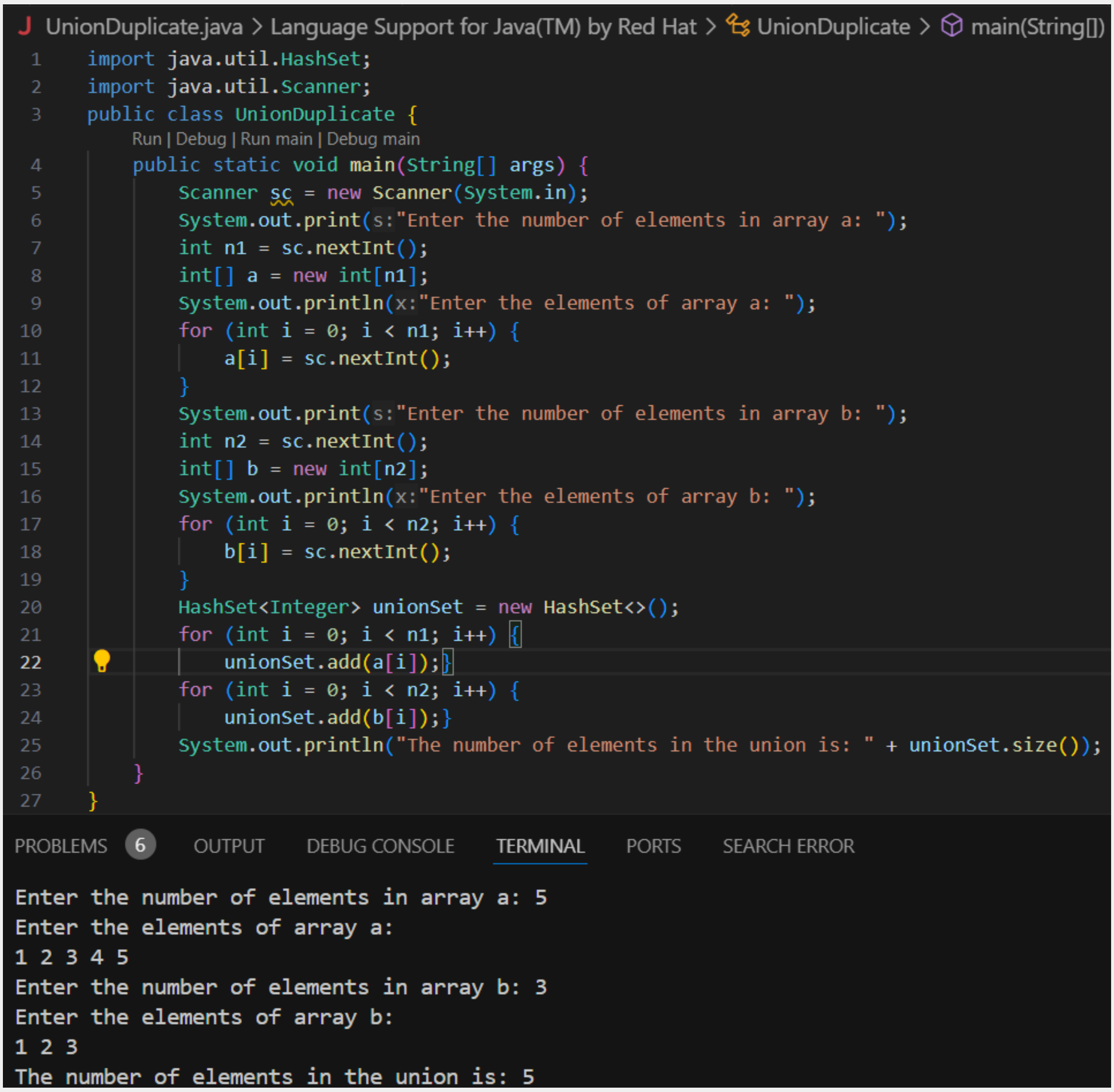
```java
static void printNGE(int arr[], int n) {
    int i = 0;
    stack s = new stack();
    s.top = -1;
    int element, next;
    s.push(arr[0]);

    for (i = 1; i < n; i++) {
        next = arr[i];

        if (s.isEmpty() == false) {
            element = s.pop();

            while (element < next) {
                System.out.println(element + " --> " + next);
                if (s.isEmpty() == true)
                    break;
                element = s.pop();
            }

            if (element > next)
                s.push(element);
        }
        s.push(next);
    }
```



```java
        while (s.isEmpty() == false) {
            element = s.pop();
            next = -1;
            System.out.println(element + " -- " + next);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter the number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println(x:"Enter the elements: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        printNGE(arr, n);
    }
}
```

```
Enter the elements:
4 5 2 25
4 --> 5
2 --> 25
5 --> 25
25 -- -1
PS D:\Coding\Java Programming\12-11 practice problem> 
```

Union of 2 arrays with duplicate elements


UnionDuplicate.java > Language Support for Java(TM) by Red Hat > UnionDuplicate > main(String[])
Run | Debug | Run main | Debug main
PROBLEMS 6
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH ERROR

```java
import java.util.HashSet;
import java.util.Scanner;
public class UnionDuplicate {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter the number of elements in array a: ");
        int n1 = sc.nextInt();
        int[] a = new int[n1];
        System.out.println(x:"Enter the elements of array a: ");
        for (int i = 0; i < n1; i++) {
            a[i] = sc.nextInt();
        }
        System.out.print(s:"Enter the number of elements in array b: ");
        int n2 = sc.nextInt();
        int[] b = new int[n2];
        System.out.println(x:"Enter the elements of array b: ");
        for (int i = 0; i < n2; i++) {
            b[i] = sc.nextInt();
        }
        HashSet<Integer> unionSet = new HashSet<>();
        for (int i = 0; i < n1; i++) {
            unionSet.add(a[i]);}
        for (int i = 0; i < n2; i++) {
            unionSet.add(b[i]);}
        System.out.println("The number of elements in the union is: " + unionSet.size());
    }
}
```

```
Enter the number of elements in array a: 5
Enter the elements of array a:
1 2 3 4 5
Enter the number of elements in array b: 3
Enter the elements of array b:
1 2 3
The number of elements in the union is: 5
```