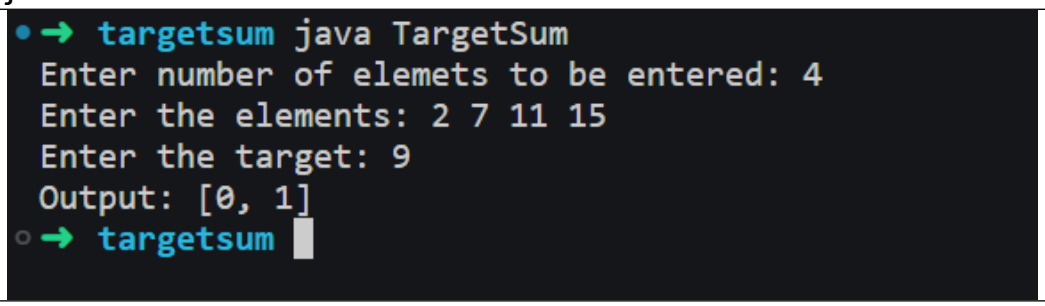
PROBLEM	Two sum
CODE	

```
import java.util.Scanner;

public class AddTwoNumbers {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Getting input of first set of numbers
        System.out.print("Enter number of digits in 1st set
of number: ");
        int size1 = sc.nextInt();
        int[] numArr1 = new int[size1];

        System.out.print("Enter 1st set of numbers: ");
        for (int i = 0; i < size1; i++) {
            // Filling up in reverse order
            numArr1[size1 - i - 1] = sc.nextInt();
        }

        // Getting input of second set of numbers
        System.out.print("Enter number of digits in 2nd set
of number: ");
        int size2 = sc.nextInt();
        int[] numArr2 = new int[size2];

        System.out.print("Enter 2nd set of numbers: ");
        for (int i = 0; i < size2; i++) {
            // Filling up in reverse order
            numArr2[size2 - i - 1] = sc.nextInt();
        }

        // ————————— SUMMING THE NUMBERS ———————
        int maxDigit = ((size1 > size2) ? size1 : size2);
        int[] sumArr = new int[maxDigit + 1]; // +1 to
manage last carry if any
        int carry = 0;
        for (int i = 0; i < sumArr.length || carry ≠ 0; i+
+) {
            int num1 = (i < size1) ? numArr1[size1 - i -
1] : 0;
            int num2 = (i < size2) ? numArr2[size2 - i -
1] : 0;

            int toBeAdded = num1 + num2 + carry;
            carry = 0;

            // Handling carry and & adding to correct unit
place
            if (toBeAdded ≥ 10) {
                sumArr[sumArr.length - i - 1] = toBeAdded %
10;
                carry += 1;
            } else {
                sumArr[sumArr.length - i - 1] = toBeAdded;
            }
        }
```

```
        // Printing inp1
        System.out.print("Inp1: [ ");
        for (int i = 0; i < size1; i++) {
            System.out.print(numArr1[size1 - i - 1] + " ");
        }
        System.out.print("]\n");

        // Printing inp2
        System.out.print("Inp2: [ ");
        for (int i = 0; i < numArr2.length; i++) {
            System.out.print(numArr2[size2 - i - 1] + " ");
        }
        System.out.print("]\n");

        // Printing output
        System.out.print("Output: ");
        for (int i = (sumArr[0] == 0) ? 1 : 0; i <
sumArr.length; i++) {
            System.out.print(sumArr[sumArr.length - i -
((sumArr[0] == 0) ? 0 : 1)] + " ");
        }
        System.out.println();
        sc.close();
    }
}
```

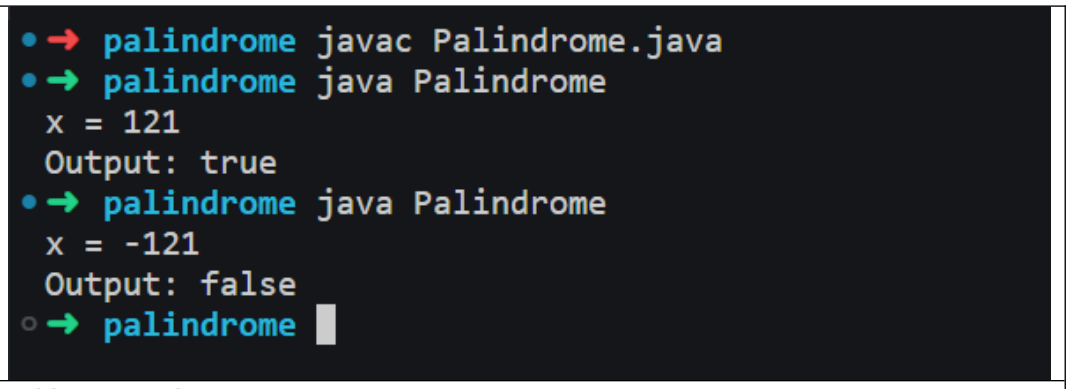
OUTPUT



PROBLEM **Palindrome Number**

CODE

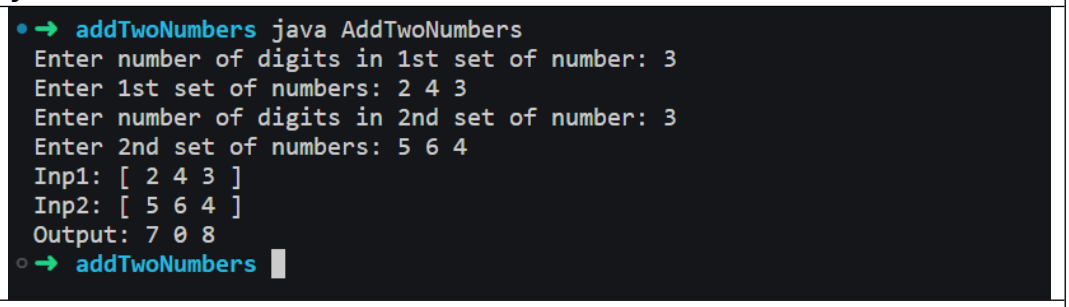
```
import java.util.Scanner;

public class Palindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("x = ");
        String inp = sc.nextLine();

        // getting the length
        boolean flag = true;
        int len = inp.length();
        for (int i = 0; i < len / 2 && flag == true; i++) {
            if (inp.charAt(i) ≠ inp.charAt(len - i - 1)) {
                flag = false;
            }
        }
        System.out.println("Output: " + flag);
    }
}
```

OUTPUT	
PROBLEM	**Add two numbers**
CODE	

```
import java.util.Scanner;

public class AddTwoNumbers {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Getting input of first set of numbers
        System.out.print("Enter number of digits in 1st set
of number: ");
        int size1 = sc.nextInt();
        int[] numArr1 = new int[size1];

        System.out.print("Enter 1st set of numbers: ");
        for (int i = 0; i < size1; i++) {
            // Filling up in reverse order
            numArr1[size1 - i - 1] = sc.nextInt();
        }

        // Getting input of second set of numbers
        System.out.print("Enter number of digits in 2nd set
of number: ");
        int size2 = sc.nextInt();
        int[] numArr2 = new int[size2];

        System.out.print("Enter 2nd set of numbers: ");
        for (int i = 0; i < size2; i++) {
            // Filling up in reverse order
            numArr2[size2 - i - 1] = sc.nextInt();
        }

        // ————————— SUMMING THE NUMBERS ————————
        int maxDigit = ((size1 > size2) ? size1 : size2);
        int[] sumArr = new int[maxDigit + 1]; // +1 to
manage last carry if any
        int carry = 0;
        for (int i = 0; i < sumArr.length || carry ≠ 0; i+
+) {
            int num1 = (i < size1) ? numArr1[size1 - i -
1] : 0;
            int num2 = (i < size2) ? numArr2[size2 - i -
1] : 0;

            int toBeAdded = num1 + num2 + carry;
            carry = 0;

            // Handling carry and & adding to correct unit
place
```

```
            if (toBeAdded ≥ 10) {
                sumArr[sumArr.length - i - 1] = toBeAdded %
10;
                carry += 1;
            } else {
                sumArr[sumArr.length - i - 1] = toBeAdded;
            }
        }

        // Printing inp1
        System.out.print("Inp1: [ ");
        for (int i = 0; i < size1; i++) {
            System.out.print(numArr1[size1 - i - 1] + " ");
        }
        System.out.print("]\n");

        // Printing inp2
        System.out.print("Inp2: [ ");
        for (int i = 0; i < numArr2.length; i++) {
            System.out.print(numArr2[size2 - i - 1] + " ");
        }
        System.out.print("]\n");

        // Printing output
        System.out.print("Output: ");
        for (int i = (sumArr[0] == 0) ? 1 : 0; i <
sumArr.length; i++) {
            System.out.print(sumArr[sumArr.length - i -
((sumArr[0] == 0) ? 0 : 1)] + " ");
        }
        System.out.println();
        sc.close();
    }
}
```

OUTPUT



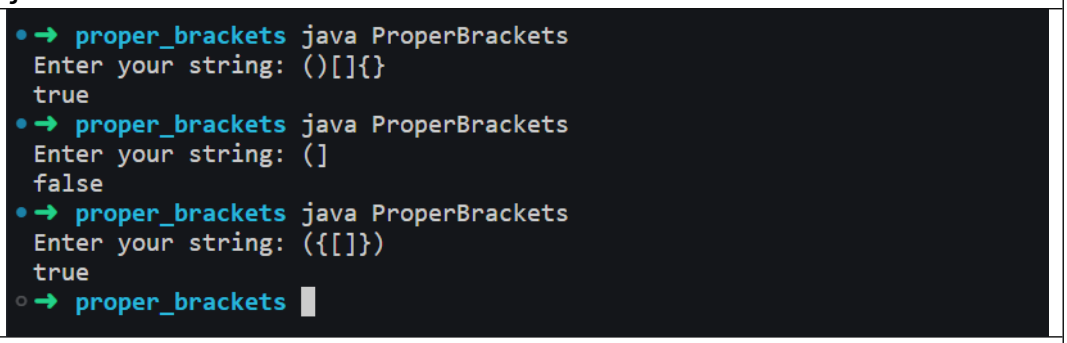
PROBLEM **Valid Parentheses**

CODE

```
import java.util.Scanner;

public class ProperBrackets {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your string: ");

        String input = sc.nextLine();

        if (input.length() % 2 ≠ 0) {
            System.out.println("false");
            return;
        }

        String tempStr = "";
```

```
        for (int i = 0; i < input.length(); i++) {
            char tempChar = input.charAt(i);
            if (tempChar == '(' || tempChar == '{' ||
tempChar == '[') {
                tempStr += tempChar;
            } else {
                if (tempChar == ')' &&
tempStr.charAt(tempStr.length() - 1) != '(') {
                    System.out.println("false");
                    return;
                } else if (tempChar == '}' &&
tempStr.charAt(tempStr.length() - 1) != '{') {
                    System.out.println("false");
                    return;
                } else if (tempChar == ']' &&
tempStr.charAt(tempStr.length() - 1) != '[') {
                    System.out.println("false");
                    return;
                } else {
                    tempStr = tempStr.substring(0,
tempStr.length() - 1);
                }
            }
        }

        System.out.println("true");
        return;
    }
}
```

OUTPUT



PROBLEM **First and last position of element in sorted array**

CODE

```
import java.util.Scanner;

public class FindFirstLast {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of inputs: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.print("Enter input array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.print("Enter target: ");
        int target = sc.nextInt();

        int start = -1;
        int end = -1;
```

```
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) {
                if (start == -1) {
                    start = i;
                }
                end = i;
            }
        }

        System.out.println("output: [" + start + ", " + end
+ "]");
    }
}
```

OUTPUT

```
➜ find_first_and_last javac FindFirstLast.java; java FindFirstLast
Enter no. of inputs: 6
Enter input array: 5 7 7 8 8 10
Enter target: 8
output: [3, 4]
➜ find_first_and_last javac FindFirstLast.java; java FindFirstLast
Enter no. of inputs: 6
Enter input array: 5 7 7 8 8 10
Enter target: 6
output: [-1, -1]
➜ find_first_and_last
```

PROBLEM

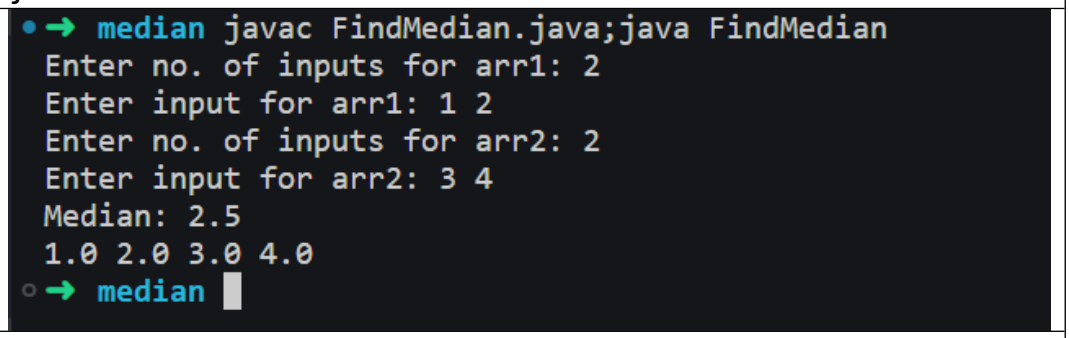
Median of two sorted array

CODE

```
import java.util.Scanner;

public class FindMedian {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter no. of inputs for arr1: ");
        int n = sc.nextInt();

        float[] arr1 = new float[n];
        System.out.print("Enter input for arr1: ");
        for (int i = 0; i < n; i++) {
            arr1[i] = sc.nextFloat();
        }

        System.out.print("Enter no. of inputs for arr2: ");
        n = sc.nextInt();

        float[] arr2 = new float[n];
        System.out.print("Enter input for arr2: ");
        for (int i = 0; i < n; i++) {
            arr2[i] = sc.nextFloat();
        }

        // Merging two arrays
        float[] newarr = new float[(arr1.length +
arr2.length)];
        int i, j, k;
        for (i = 0, j = 0, k = 0; i < arr1.length && j <
arr2.length; k++) {
            if (arr1[i] > arr2[j]) {
                newarr[k] = arr2[j];
                j++;
            } else {
                newarr[k] = arr1[i];
```

```
                i++;
            }
        }

        while (i < arr1.length) {
            newarr[k] = arr1[i];
            i++;
            k++;
        }
        while (j < arr2.length) {
            newarr[k] = arr2[j];
            j++;
            k++;
        }

        // find median
        if (newarr.length % 2 == 0) {
            // In case of even →
            float result = (newarr[(newarr.length - 1) / 2]
+ newarr[((newarr.length - 1) / 2) + 1]) / 2;
            System.out.println("Median: " + result);
        } else {
            // In case of odd →
            float result = (newarr[(newarr.length - 1) / 2])
/ 2;
            System.out.println("Median: " + result);
        }

        for (int m = 0; m < newarr.length; m++) {
            System.out.print(newarr[m] + " ");
        }
        System.out.println();
    }
}
```

OUTPUT



PROBLEM **First missing positive**

CODE

```
import java.util.Scanner;

public class FirstMissingPositive {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of inputs: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.print("Enter input array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        for (int i = 1; i ≤ arr.length; i++) {
```

```
                boolean found = false;

                for (int j = 0; j < arr.length && !found; j++) {
                    if (arr[j] == i) {
                        found = true;
                    }
                }
                if (!found) {
                    System.out.println(i);
                    return;
                }
            }
        }

}
```

OUTPUT