Problem 1: Write a Java program to find the sum of all odd numbers in an array.

Code:

Program1.java | Program2.java | Program3.java | Program4.java | Program5.java | Pro

Source | History

```java
// 1. Write a Java program to find the sum of all odd numbers in an array.
//
package com.mycompany.programs;

import java.util.Scanner;

public class Program1 {

    public static void main(String[] args) {
        Scanner inp = new Scanner(source: System.in);

        System.out.print(s: "Enter the size of an array : ");
        int size = inp.nextInt();

        int[] array = new int[size];

        for (int i = 0; i < array.length; i++) {
            System.out.printf(format: "Enter element - %d : ", (i + 1));
            array[i] = inp.nextInt();
        }

        int sum = 0;

        for (int i = 0; i < array.length; i++) {
            if (array[i] % 2 != 0) {
                sum += array[i];
            }
        }

        System.out.println("Sum of all odd numbers : " + sum);

    }
}
```

Output:

```
--- exec:3.1.0:exec (default-cli) @ 01Fi
Enter the size of the array : 5
Enter element - 1 : 3
Enter element - 2 : 14
Enter element - 3 : 15
Enter element - 4 : 81
Enter element - 5 : 50
Element - 1 : 3
Element - 2 : 14
Element - 3 : 15
Element - 4 : 81
Element - 5 : 50
------------------------------------------
BUILD SUCCESS
------------------------------------------
Total time:  15.373 s
Finished at: 2025-02-07T15:24:00+06:00
------------------------------------------
```

Problem 2: Write a Java program that takes two arrays as input, calculate the index wise sum of these arrays, and store the result in a third array.

Code:

```java
/*
2. Write a Java program that takes two arrays as input, calculate the
index wise sum of these arrays, and store the result in a third array.
 */
package com.mycompany.programs;

import java.util.Scanner;

public class Program2 {

    public static void main(String[] args) {
        Scanner inp = new Scanner(source: System.in);

        System.out.print(s: "Enter the size of both Arrays : ");
        int size = inp.nextInt();

        int[] array1 = new int[size];

        System.out.println(x: "Array1 elements input : ");
        for (int i = 0; i < array1.length; i++) {
            System.out.printf(format: "Enter element - %d : ", (i + 1));
            array1[i] = inp.nextInt();
        }

        int[] array2 = new int[size];

        System.out.println(x: "\nArray2 elements input : ");
        for (int i = 0; i < array2.length; i++) {
            System.out.printf(format: "Enter element - %d : ", (i + 1));
            array2[i] = inp.nextInt();
        }

        int[] array3 = new int[size];

        for (int i = 0; i < size; i++) {
            array3[i] = array1[i] + array2[i];
        }

        System.out.println(x: "Index wise summed Array : ");
        for (int i = 0; i < size; i++) {
            System.out.printf(format: "%d ", array3[i]);
        }
    }
}
```

Output:

```
Output - Run (Program2)
Enter the size of both Arrays : 7
Array1 elements input :
Enter element - 1 : 1
Enter element - 2 : 2
Enter element - 3 : 3
Enter element - 4 : 4
Enter element - 5 : 5
Enter element - 6 : 6
Enter element - 7 : 7

Array2 elements input :
Enter element - 1 : 8
Enter element - 2 : 9
Enter element - 3 : 10
Enter element - 4 : 11
Enter element - 5 : 12
Enter element - 6 : 13
Enter element - 7 : 14
Index wise summed Array :
9 11 13 15 17 19 21
------------------------------------
BUILD SUCCESS
```

Problem 3: Write a Java program to search an element in an array.

Code:

```java
/*
3. Write a Java program to search an element in an array.
 */
package com.mycompany.programs;

import java.util.Scanner;

public class Program3 {

    public static void main(String[] args) {
        Scanner inp = new Scanner(source:  System.in);

        System.out.print(s: "Enter the size of an array : ");
        int size = inp.nextInt();

        int[] array = new int[size];

        for (int i = 0; i < array.length; i++) {
            System.out.printf(format:  "Enter element - %d : ", (i + 1));
            array[i] = inp.nextInt();
        }

        System.out.println(x: "Enter element you want to search : ");
        int search = inp.nextInt();

        boolean result = false;

        for (int i = 0; i < array.length; i++) {
            if (array[i] == search) {
                System.out.printf(format:  "Element found at index %d", args:  i);
                result = true;
                break;
            }
        }

        if (!result) {
            System.out.println(x: "Error! element not found");
        }
    }
}
```

Output:

```
Output - Run (Program3)
--- exec:3.1.0:exec (default-cli) @
Enter the size of an array : 5
Enter element - 1 : 65
Enter element - 2 : 47
Enter element - 3 : 91
Enter element - 4 : 35
Enter element - 5 : 88
Enter element you want to search :
91
Element found at index 2
------------------------------------
```

```
Output - Run (Program3)
--- exec:3.1.0:exec (default-cli) @ L
Enter the size of an array : 5
Enter element - 1 : 94
Enter element - 2 : 65
Enter element - 3 : 74
Enter element - 4 : 33
Enter element - 5 : 28
Enter element you want to search :
30
Error! element not found
------------------------------------
```

Problem 4: Write a Java program to reverse the elements in an array without using a second array.

Code:

```
Program1.java × Program2.java × Program3.java × Program4.java × Program5.java ×
Source History
```

```java
/*
4. Write a Java program to reverse the elements
in an array without using a second array.
*/
package com.mycompany.programs;

import java.util.Scanner;

public class Program4 {

    public static void main(String[] args) {
        Scanner inp = new Scanner(source: System.in);

        System.out.print(s: "Enter the size of an array : ");
        int size = inp.nextInt();

        int[] array = new int[size];

        for (int i = 0; i < array.length; i++) {
            System.out.printf(format: "Enter element - %d : ", (i + 1));
            array[i] = inp.nextInt();
        }

        System.out.print(s: "Array elements : ");
        for (int i = 0; i < size; i++) {
            System.out.printf(format: "%d ", array[i]);
        }

        int temp;

        for (int i = 0; i < size / 2; i++) {
            temp = array[i];
            array[i] = array[size - 1 - i];
            array[size - 1 - i] = temp;
        }

        System.out.println(x: "Reversed array : ");
        for (int i = 0; i < size; i++) {
            System.out.printf(format: "%d ", array[i]);
        }
    }
}
```

Output:

```
Output - Run (Program4) ×
    --release 22 is recommended instead of

--- exec:3.1.0:exec (default-cli) @ LabAss
Enter the size of an array : 6
Enter element - 1 : 14
Enter element - 2 : 18
Enter element - 3 : 78
Enter element - 4 : 93
Enter element - 5 : 51
Enter element - 6 : 62
Array elements : 14 18 78 93 51 62
Reversed array :
62 51 93 78 18 14
------------------------------------------------------------------------
BUILD SUCCESS
```

Problem 5: Write a Java program to find the second highest element of an array.

Code:

```java
/*
5. Write a Java program to find the second highest eler
 */
package com.mycompany.programs;

import java.util.Scanner;

public class Program5 {

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

        System.out.println(1 != 15);

        System.out.print("Enter the size of an array : ");
        int size = inp.nextInt();

        int[] array = new int[size];

        for (int i = 0; i < array.length; i++) {
            System.out.printf("Enter element - %d : ", (i + 1));
            array[i] = inp.nextInt();
        }
        System.out.print("Array elements : ");
        for (int i = 0; i < size; i++) {
            System.out.printf("%d ", array[i]);
        }

        int max = array[0];
        int secondMax = array[0];

        for (int i = 0; i < array.length; i++) {
            if (array[i] > max) {
                secondMax = max;
                max = array[i];
            } else if (array[i] > secondMax && array[i] != max) {
                secondMax = array[i];
            }
        }

        System.out.println("\nSecond highest element: " + secondMax);

        inp.close();
    }
}
```

Output:

```
Output - Run (Program4)
    --release 22 is recommended instead of

--- exec:3.1.0:exec (default-cli) @ LabAss
Enter the size of an array : 6
Enter element - 1 : 14
Enter element - 2 : 18
Enter element - 3 : 78
Enter element - 4 : 93
Enter element - 5 : 51
Enter element - 6 : 62
Array elements : 14 18 78 93 51 62
Reversed array :
62 51 93 78 18 14
------------------------------------------
BUILD SUCCESS
```

Problem 6: Write a Java program that calculates the average of an array, excluding the highest and lowest values in the array.

Code:

```java
/*
6. Write a Java program that calculates the average of an
array, excluding the highest and lowest values in the array.
 */
package com.mycompany.programs;

import java.util.Scanner;

public class Program6 {

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

        System.out.print("Enter the size of an array : ");
        int size = inp.nextInt();

        int[] array = new int[size];

        for (int i = 0; i < array.length; i++) {
            System.out.printf("Enter element - %d : ", (i + 1));
            array[i] = inp.nextInt();
        }

        int max = array[0];
        int min = array[0];
        int sum = 0;

        for (int i = 0; i < array.length; i++) {
            if (array[i] > max) {
                max = array[i];
            }
            if (array[i] < min) {
                min = array[i];
            }
        }

        for (int i = 0; i < array.length; i++) {
            if (array[i] != max && array[i] != min) {
                sum += array[i];
            }
        }

        System.out.println("Sum of the array : " + sum);

        double average = sum / (array.length - 2);
        System.out.println("Average : " + average);
    }
}
```

Output:

```
Output - Run (Program6)
--- exec:3.1.0:exec (default-cli) @ L
Enter the size of an array : 6
Enter element - 1 : 10
Enter element - 2 : 20
Enter element - 3 : 30
Enter element - 4 : 40
Enter element - 5 : 50
Enter element - 6 : 60
Sum of the array : 140
Average : 35.0
------------------------------------
BUILD SUCCESS
------------------------------------
Total time:  27.600 s
```

Problem 7: Write a Java program to calculate the sum of all the prime numbers in a 2D array.

Code:

```java
/*7. Write a Java program to calculate the sum of all the prime numbers in a 2D array. */
package com.mycompany.programs;

import java.util.Scanner;

public class Program7 {

    public static boolean isPrime(int n) {
        if (n < 2) {
            return false;
        }

        int i = 2;
        while (i * i <= n) {
            if (n % i == 0) {
                return false;
            }
            i++;
        }
        return true;
    }

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

        int row, column;

        System.out.print("Enter row in array: ");
        row = sc.nextInt();

        System.out.print("Enter column in array: ");
        column = sc.nextInt();

        int[][] array = new int[row][column];

        for (int i = 0; i < row; i++) {
            System.out.printf("Input %d elements for row %d: ", row, i + 1);

            for (int j = 0; j < column; j++) {
                array[i][j] = sc.nextInt();
            }
        }

        int sum = 0;
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < column; j++) {
                if (isPrime(array[i][j])) {
                    sum += array[i][j];
                }
            }
        }
        System.out.println("Sum of prime numbers: " + sum);
    }
}
```

Output:

```
Output - Run (Program7)
    --release 22 is recommended inste
lly

--- exec:3.1.0:exec (default-cli) @ L
Enter row in array: 3
Enter column in array: 3
Input 3 elements for row 1: 3 2 1
Input 3 elements for row 2: 6 5 4
Input 3 elements for row 3: 9 8 7
Sum of prime numbers: 17
------------------------------------
BUILD SUCCESS
------------------------------------
Total time:  01:36 min
```