



```
1* import java.util.ArrayList;
2 import java.util.Collections;
3
4* public class MergeSortedArrays {
5*     public static void main(String[] args) {
6         int[] arr1 = {1, 3, 5};
7         int[] arr2 = {2, 4, 6};
8         ArrayList<Integer> list = new ArrayList<>();
9
10*        for (int i : arr1) {
11            list.add(i);
12        }
13*        for (int i : arr2) {
14            list.add(i);
15        }
16
17        Collections.sort(list);
18        System.out.println(list);
19    }
20 }
```

```
java -cp /tmp/6q5IKNylQN/MergeSortedArrays
[1, 2, 3, 4, 5, 6]
```

```
=== Code Execution Successful ===
```

n.java



Run

Output

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

```
public class Statistics {
```

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

```
        int[] numbers = {1, 2, 3, 4, 5, 5, 5};
```

```
        Arrays.sort(numbers);
```

```
        double mean = getMean(numbers);
```

```
        double median = getMedian(numbers);
```

```
        int mode = getMode(numbers);
```

```
        System.out.println("Mean: " + mean);
```

```
        System.out.println("Median: " + median);
```

```
        System.out.println("Mode: " + mode);
```

```
    }
```

```
    private static double getMean(int[] numbers) {
```

```
        int sum = 0;
```

```
        for (int num : numbers) {
```

```
            sum += num;
```

```
        }
```

```
        return (double) sum / numbers.length;
```

```
    }
```

```
    private static double getMedian(int[] numbers) {
```

```
        int middle = numbers.length / 2;
```

```
java -cp /tmp/D934YNsZPa/Statistics
```

```
Mean: 3.5714285714285716
```

```
Median: 4.0
```

```
Mode: 5
```

```
=== Code Execution Successful ===
```

Main.java



Run

Output

```

1 public class CompositeNumbers {
2     public static void main(String[] args) {
3         int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
4         int count = 0;
5         for (int num : numbers) {
6             if (isComposite(num)) {
7                 count++;
8             }
9         }
10        System.out.println("Number of composite numbers: " + count);
11    }
12
13    private static boolean isComposite(int num) {
14        if (num <= 1) {
15            return false;
16        }
17        for (int i = 2; i * i <= num; i++) {
18            if (num % i == 0) {
19                return true;
20            }
21        }
22        return false;
23    }

```

```

^ java -cp /tmp/pdDntaxZcx/CompositeNumbers
Number of composite numbers: 5

=== Code Execution Successful ===

```



Main.java



Run

Output

```
1- public class MatrixMultiplication {  
2-     public static void main(String[] args) {  
3         int[][] matrixA = {{1, 2, 3}, {4, 5, 6}};  
4         int[][] matrixB = {{7, 8}, {9, 10}, {11, 12}};  
5         int[][] result = new int[matrixA.length][matrixB[0].length];  
6  
7-         for (int i = 0; i < matrixA.length; i++) {  
8-             for (int j = 0; j < matrixB[0].length; j++) {  
9-                 for (int k = 0; k < matrixA[0].length; k++) {  
10                    result[i][j] += matrixA[i][k] * matrixB[k][j];  
11                }  
12            }  
13        }  
14  
15-         for (int i = 0; i < result.length; i++) {  
16-             for (int j = 0; j < result[0].length; j++) {  
17                 System.out.print(result[i][j] + " ");  
18             }  
19             System.out.println();  
20         }  
21     }  
22 }
```

```
java -cp /tmp/Qt7BbMwohp/MatrixMultiplication  
58 64  
139 154  
  
=== Code Execution Successful ===
```

Main.java



Run

Output

```
1- public class MatrixAddition {
2-     public static void main(String[] args) {
3-         int[][] matrixA = {{1, 2}, {3, 4}};
4-         int[][] matrixB = {{5, 6}, {7, 8}};
5-         int[][] result = new int[matrixA.length][matrixA[0].length];
6-
7-         for (int i = 0; i < matrixA.length; i++) {
8-             for (int j = 0; j < matrixA[0].length; j++) {
9-                 result[i][j] = matrixA[i][j] + matrixB[i][j];
10-            }
11-        }
12-
13-        for (int i = 0; i < result.length; i++) {
14-            for (int j = 0; j < result[0].length; j++) {
15-                System.out.print(result[i][j] + " ");
16-            }
17-            System.out.println();
18-        }
19-    }
20- }
```

```
java -cp /tmp/09m5vbldh0/MatrixAddition
```

```
6 8
```

```
10 12
```

```
=== Code Execution Successful ===
```