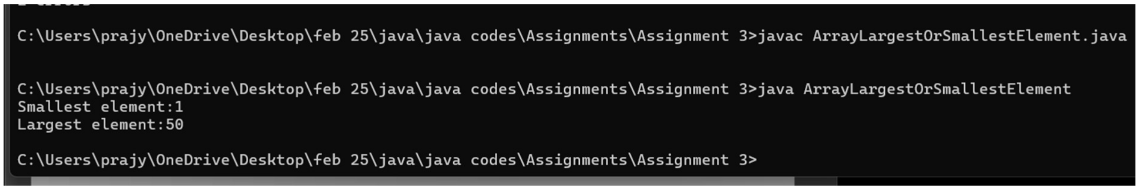
Assignment no 3

Array coding question :

1.Find the Largest and Smallest Element ○ Given an array, find the smallest and largest elements in it.

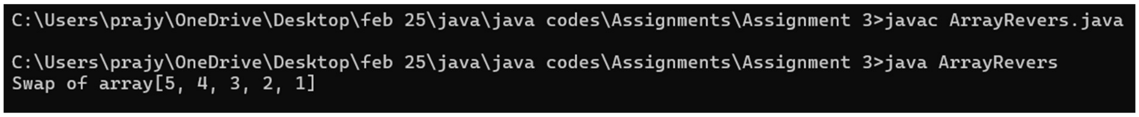
```
import java.util.Scanner;


class ArrayLargestOrSmallestElement{


    public static void main(String args[]){


        int arr[] = {10,2,1,40,50,15,20};


        int min = arr[0];
        int max = arr[0];


        for(int i=0;i<arr.length;i++)
        {
            if (arr[i] < min){
                min = arr[i];
            }else if(arr[i] > max){
                max = arr[i];
            }
        }
            System.out.println("Smallest element:" + min);
            System.out.println("Largest element:"+ max);


    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac ArrayLargestOrSmallestElement.java


C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java ArrayLargestOrSmallestElement
Smallest element:1
Largest element:50

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>
```

2.Reverse an Array ○ Reverse the given array in place.

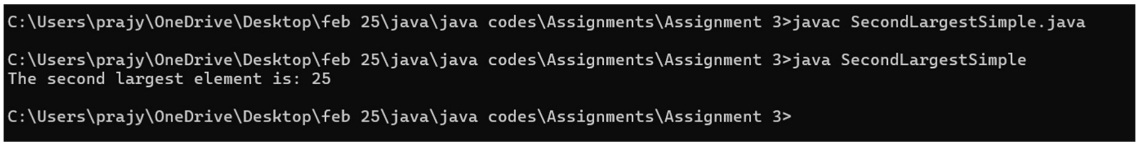
```
import java.util.Scanner;

import java.util.Arrays;

class ArrayRevers{


        public static void main(String args[]){


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


    // Swap elements from start to end

    for (int i = 0; i < arr.length / 2; i++) {

      int t = arr[i];

      arr[i] = arr[arr.length - 1 - i];

      arr[arr.length - 1 - i] = t;

    }


    System.out.println("Swap of array" + Arrays.toString(arr));

  }

}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac ArrayRevers.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java ArrayRevers
Swap of array[5, 4, 3, 2, 1]
```

2Find the Second Largest Element ○ Find the second-largest element in the given array.

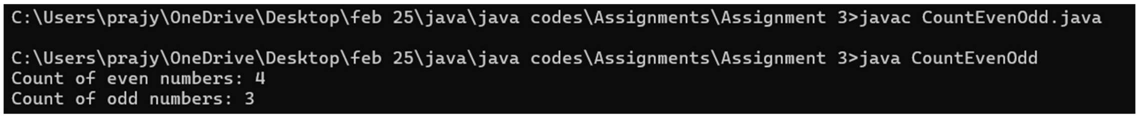
```
public class SecondLargestSimple {
    public static void main(String[] args) {
        int[] arr = {10, 20, 5, 30, 25};
        if (arr.length < 2) {
            System.out.println("Array must have at least two elements.");
            return;
        }
        int first = arr[0];
        int second = Integer.MIN_VALUE;
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] > first) {
                second = first;
                first = arr[i]; // Update largest
            } else if (arr[i] > second && arr[i] != first) {
                second = arr[i]; // Update second largest
            }
        }
        if (second == Integer.MIN_VALUE) {
            System.out.println("No second largest element found.");
        } else {
            System.out.println("The second largest element is: " + second);
        }
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac SecondLargestSimple.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java SecondLargestSimple
The second largest element is: 25

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>
```

4. Count Even and Odd Numbers ○ Count the number of even and odd numbers in an array.

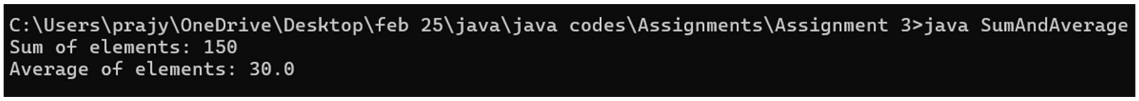
```
public class CountEvenOdd {
    public static void main(String[] args) {
        int[] arr = {10, 21, 32, 43, 54, 65, 76};
        int evenCount = 0, oddCount = 0;

        for (int num : arr) {
            if (num % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
        }

        System.out.println("Count of even numbers: " + evenCount);
        System.out.println("Count of odd numbers: " + oddCount);
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac CountEvenOdd.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java CountEvenOdd
Count of even numbers: 4
Count of odd numbers: 3
```

5. Find Sum and Average ○ Compute the sum and average of all elements in the array.

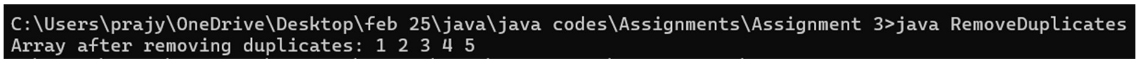
```
public class SumAndAverage {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50};


        int sum = 0;
        for (int num : arr) {
            sum += num;
        }
        double average = (double) sum / arr.length;

        System.out.println("Sum of elements: " + sum);
        System.out.println("Average of elements: " + average);
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java SumAndAverage
Sum of elements: 150
Average of elements: 30.0
```

6. Remove Duplicates from a Sorted Array ○ Remove duplicate elements from a sorted array without using extra space.

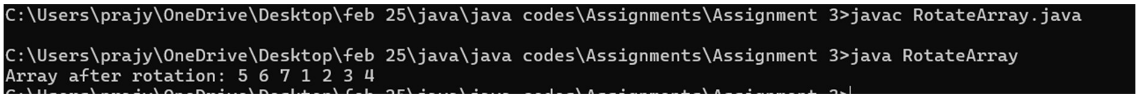
```
public class RemoveDuplicates {
    public static int removeDuplicates(int[] arr) {
        if (arr.length == 0) return 0;

        int uniqueIndex = 0;
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] != arr[uniqueIndex]) {
                uniqueIndex++;
                arr[uniqueIndex] = arr[i];
            }
        }
        return uniqueIndex + 1;
    }
    public static void main(String[] args) {
        int[] arr = {1, 1, 2, 2, 3, 4, 4, 5};
        int newLength = removeDuplicates(arr);

        System.out.print("Array after removing duplicates: ");
        for (int i = 0; i < newLength; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java RemoveDuplicates
Array after removing duplicates: 1 2 3 4 5
```

7. Rotate an Array ○ Rotate the array to the right by k positions.

```
public class RotateArray {
    public static void rotate(int[] arr, int k) {
        int n = arr.length;
        k = k % n;
        reverse(arr, 0, n - 1);
        reverse(arr, 0, k - 1);
        reverse(arr, k, n - 1);
    }
    private static void reverse(int[] arr, int start, int end) {
        while (start < end) {
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5, 6, 7};
        int k = 3; // Rotate by 3 positions

        rotate(arr, k);

        System.out.print("Array after rotation: ");
        for (int num : arr) {
            System.out.print(num + " ");

        }

    }

}
```

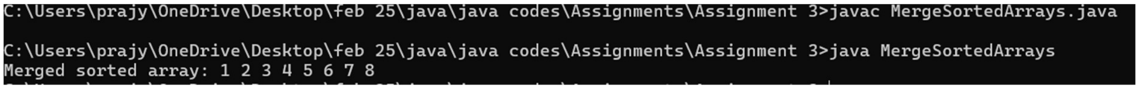


```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac RotateArray.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java RotateArray
Array after rotation: 5 6 7 1 2 3 4
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>
```

8. Merge Two Sorted Arrays ○ Merge two sorted arrays into a single sorted array without using extra space.

```
public class MergeSortedArrays {

    public static void mergeSortedArrays(int[] arr1, int[] arr2) {

        int m = arr1.length;

        int n = arr2.length;

        int i = m - 1, j = n - 1, k = m + n - 1;



        int[] mergedArray = new int[m + n];

        System.arraycopy(arr1, 0, mergedArray, 0, m);



        while (i >= 0 && j >= 0) {

            if (arr1[i] > arr2[j]) {

                mergedArray[k--] = arr1[i--];
            } else {
                mergedArray[k--] = arr2[j--];
            }
        }

        while (j >= 0) {
            mergedArray[k--] = arr2[j--];
        }

        System.out.print("Merged sorted array: ");
        for (int num : mergedArray) {
            System.out.print(num + " ");
        }
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 3, 5, 7};
        int[] arr2 = {2, 4, 6, 8};

        mergeSortedArrays(arr1, arr2);
```

```
  }

}
```

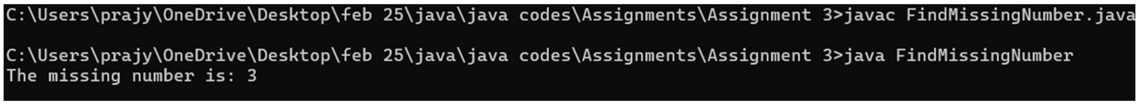


```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac MergeSortedArrays.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java MergeSortedArrays
Merged sorted array: 1 2 3 4 5 6 7 8
```

9. Find Missing Number in an Array ○ Given an array of size n-1 containing numbers from 1 to n, find the missing number.

```
public class FindMissingNumber {

  public static int findMissingNumber(int[] arr, int n) {

    int totalSum = n * (n + 1) / 2;

    int arrSum = 0;



    for (int num : arr) {

      arrSum += num;

    }



    return totalSum - arrSum;

  }



  public static void main(String[] args) {

    int[] arr = {1, 2, 4, 5, 6};
```

```
        int n = 6;



        int missingNumber = findMissingNumber(arr, n);

        System.out.println("The missing number is: " + missingNumber);

    }

}
```



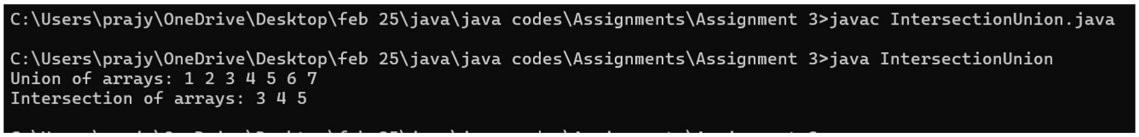
```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac FindMissingNumber.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java FindMissingNumber
The missing number is: 3
```

10. Find Intersection and Union of Two Arrays ○ Find the intersection and union of two unsorted arrays.

```
public class IntersectionUnion {

    public static void main(String[] args) {

        int[] arr1 = {1, 2, 3, 4, 5}; // 1 array

        int[] arr2 = {3, 4, 5, 6, 7}; // 2 array



        System.out.print("Union of arrays: ");

        for (int num : arr1) {

            System.out.print(num + " ");

        }

        for (int num : arr2) {
```

```
    boolean isDuplicate = false;
    for (int val : arr1) {
        if (num == val) {
            isDuplicate = true;
            break;
        }
    }
    if (!isDuplicate) {
        System.out.print(num + " ");
    }
}
System.out.println();

System.out.print("Intersection of arrays: ");
for (int num : arr1) {
    for (int val : arr2) {
        if (num == val) {
            System.out.print(num + " ");
            break;
        }
```

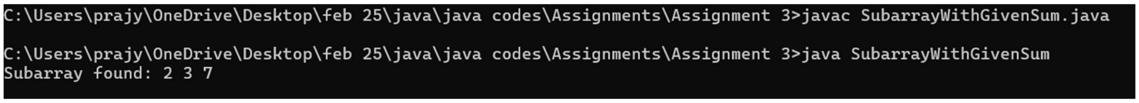
```
            }
        }
        System.out.println();
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac IntersectionUnion.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java IntersectionUnion
Union of arrays: 1 2 3 4 5 6 7
Intersection of arrays: 3 4 5
```

11. Find a Subarray with Given Sum ○ Given an array of integers, find the subarray that sums to a given value S.

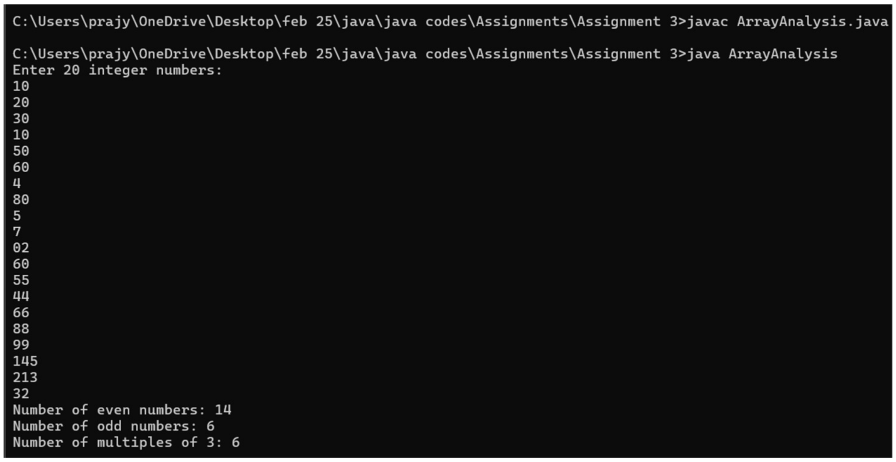
```
public class SubarrayWithGivenSum {
    public static void findSubarray(int[] arr, int S) {
        int n = arr.length;
        for (int i = 0; i < n; i++) {
            int sum = 0;
            for (int j = i; j < n; j++) {
                sum += arr[j];
                if (sum == S) {
                    System.out.print("Subarray found: ");
                    for (int k = i; k <= j; k++) {
                        System.out.print(arr[k] + " ");
                    }
                    System.out.println();
                    return;
                }
            }
        }
        System.out.println("No subarray found with sum " + S);
    }

    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 7, 5};
        int S = 12;

        findSubarray(arr, S);
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac SubarrayWithGivenSum.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java SubarrayWithGivenSum
Subarray found: 2 3 7
```

12. Write a program to accept 20 integer numbers in a single Dimensional Array. Find and Display the following: ○ Number of even numbers. ○ Number of odd numbers. ○ Number of multiples of 3

```
import java.util.Scanner;

public class ArrayAnalysis {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] arr = new int[20];
        int evenCount = 0, oddCount = 0, multipleOfThreeCount = 0;

        System.out.println("Enter 20 integer numbers:");
        for (int i = 0; i < 20; i++) {
            arr[i] = scanner.nextInt();

            if (arr[i] % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
            if (arr[i] % 3 == 0) {

                multipleOfThreeCount++;

            }

        }



        System.out.println("Number of even numbers: " + evenCount);

        System.out.println("Number of odd numbers: " + oddCount);

        System.out.println("Number of multiples of 3: " + multipleOfThreeCount);



    }

}
```

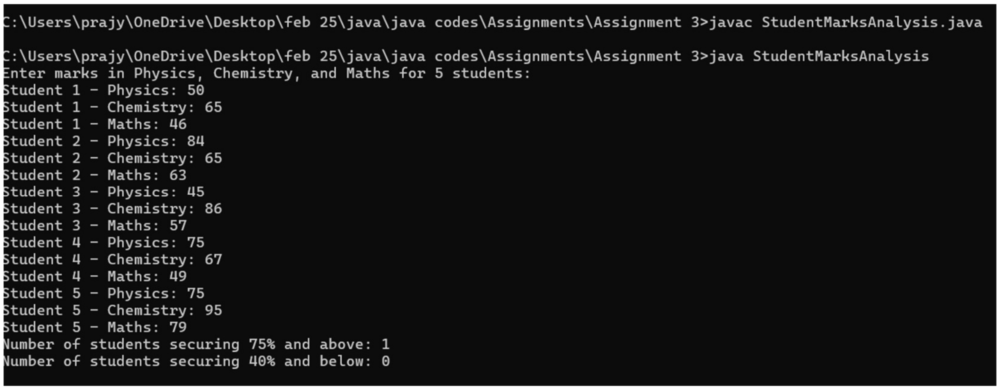


```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac ArrayAnalysis.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java ArrayAnalysis
Enter 20 integer numbers:
10
20
30
10
50
60
4
80
5
7
02
60
55
44
66
88
99
145
213
32
Number of even numbers: 14
Number of odd numbers: 6
Number of multiples of 3: 6
```

13. Write a program to accept the marks in Physics, Chemistry and Maths secured by 20 class students in a single Dimensional Array. Find and display the following: ○ Number of students securing 75% and above in aggregate. ○ Number of students securing 40% and below in aggregate

```
import java.util.Scanner;

public class StudentMarksAnalysis {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int numStudents = 5;
    double[] marks = new double[numStudents];
    int above75Count = 0, below40Count = 0;

    System.out.println("Enter marks in Physics, Chemistry, and Maths for " + numStudents + " students:");

    for (int i = 0; i < numStudents; i++) {
      System.out.print("Student " + (i + 1) + " - Physics: ");
      double physics = scanner.nextDouble();
      System.out.print("Student " + (i + 1) + " - Chemistry: ");
      double chemistry = scanner.nextDouble();
```

```
            System.out.print("Student " + (i + 1) + " - Maths: ");
            double maths = scanner.nextDouble();

            double aggregate = (physics + chemistry + maths) / 3;
            marks[i] = aggregate;

            if (aggregate >= 75) {
                above75Count++;
            } else if (aggregate <= 40) {
                below40Count++;
            }
        }

        System.out.println("Number of students securing 75% and above: " + above75Count);
        System.out.println("Number of students securing 40% and below: " + below40Count);

        scanner.close();
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac StudentMarksAnalysis.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java StudentMarksAnalysis
Enter marks in Physics, Chemistry, and Maths for 5 students:
Student 1 - Physics: 50
Student 1 - Chemistry: 65
Student 1 - Maths: 46
Student 2 - Physics: 84
Student 2 - Chemistry: 65
Student 2 - Maths: 63
Student 3 - Physics: 45
Student 3 - Chemistry: 86
Student 3 - Maths: 57
Student 4 - Physics: 75
Student 4 - Chemistry: 67
Student 4 - Maths: 49
Student 5 - Physics: 75
Student 5 - Chemistry: 95
Student 5 - Maths: 79
Number of students securing 75% and above: 1
Number of students securing 40% and below: 0
```

15. Write a Java program to print all sub-arrays with 0 sum present in a given array of integers.

Example: Input :

nums1 = { 1, 3, -7, 3, 2, 3, 1, -3, -2, -2 }

nums2 = { 1, 2, -3, 4, 5, 6 }

nums3= { 1, 2, -2, 3, 4, 5, 6 }

Output:

Sub-arrays with 0 sum : [1, 3, -7, 3]

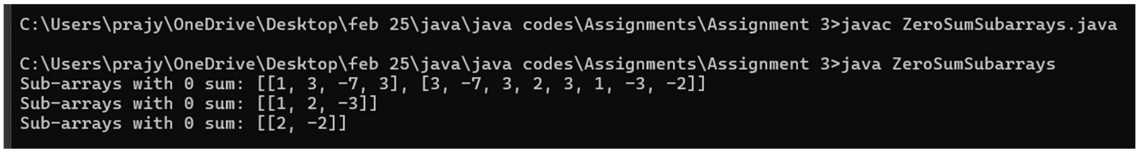
Sub-arrays with 0 sum : [3, -7, 3, 2, 3, 1, -3, -2]

Sub-arrays with 0 sum : [1, 2, -3]

Sub-arrays with 0 sum : [2, -2]

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;
```

```
public class ZeroSumSubarrays {
    public static void findZeroSumSubarrays(int[] arr) {
        HashMap<Integer, List<Integer>> map = new HashMap<>();
        List<List<Integer>> result = new ArrayList<>();
        int sum = 0;

        map.put(0, new ArrayList<>());
        map.get(0).add(-1);

        for (int i = 0; i < arr.length; i++) {
            sum += arr[i];

            if (map.containsKey(sum)) {
                for (int start : map.get(sum)) {
                    List<Integer> subarray = new ArrayList<>();
                    for (int j = start + 1; j <= i; j++) {
                        subarray.add(arr[j]);
                    }
                    result.add(subarray);
                }
            }

            map.putIfAbsent(sum, new ArrayList<>());
            map.get(sum).add(i);
        }

        System.out.println("Sub-arrays with 0 sum: " + result);
    }

    public static void main(String[] args) {
        int[] nums1 = {1, 3, -7, 3, 2, 3, 1, -3, -2, -2};
        int[] nums2 = {1, 2, -3, 4, 5, 6};
        int[] nums3 = {1, 2, -2, 3, 4, 5, 6};

        findZeroSumSubarrays(nums1);
        findZeroSumSubarrays(nums2);
        findZeroSumSubarrays(nums3);
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>javac ZeroSumSubarrays.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 3>java ZeroSumSubarrays
Sub-arrays with 0 sum: [[1, 3, -7, 3], [3, -7, 3, 2, 3, 1, -3, -2]]
Sub-arrays with 0 sum: [[1, 2, -3]]
Sub-arrays with 0 sum: [[2, -2]]
```

16. Given two sorted arrays A and B of size p and q, write a Java program to merge elements of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements.

Example: Input :

int[] A = { 1, 5, 6, 7, 8, 10 }

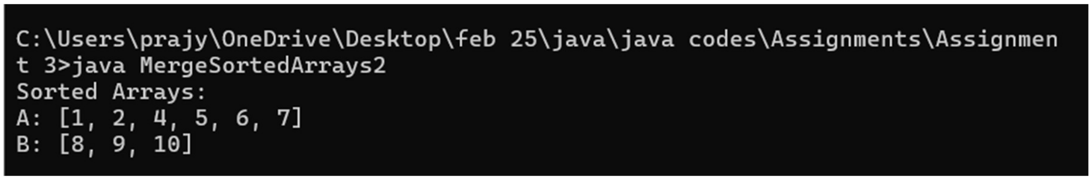
int[] B = { 2, 4, 9 }

Output:

Sorted Arrays: A: [1, 2, 4, 5, 6, 7]

B: [8, 9, 10]

```
import java.util.Arrays;



public class MergeSortedArrays {

    public static void mergeSortedArrays(int[] A, int[] B) {

        int p = A.length;

        int q = B.length;

        int[] merged = new int[p + q];
        System.arraycopy(A, 0, merged, 0, p);
        System.arraycopy(B, 0, merged, p, q);

        Arrays.sort(merged);
        System.arraycopy(merged, 0, A, 0, p);
        System.arraycopy(merged, p, B, 0, q);
        System.out.println("Sorted Arrays:");
        System.out.println("A: " + Arrays.toString(A));
        System.out.println("B: " + Arrays.toString(B));
    }
    public static void main(String[] args) {
        int[] A = {1, 5, 6, 7, 8, 10};
        int[] B = {2, 4, 9};

        mergeSortedArrays(A, B);
    }
}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>java MergeSortedArrays2
Sorted Arrays:
A: [1, 2, 4, 5, 6, 7]
B: [8, 9, 10]
```

17. Write a Java program to find the maximum product of two integers in a given array of integers.

Example: Input :

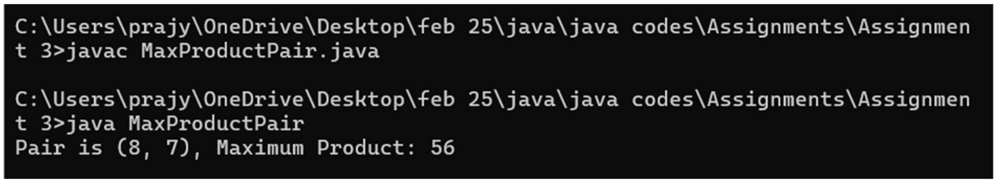
nums = { 2, 3, 5, 7, -7, 5, 8, -5 }

Output:

Pair is (7, 8), Maximum Product: 56

```
public class MaxProductPair {
    public static void findMaxProduct(int[] nums) {
        if (nums.length < 2) {
            System.out.println("Array should have at least two numbers.");
            return;
        }

        int max1 = Integer.MIN_VALUE, max2 = Integer.MIN_VALUE;
        int min1 = Integer.MAX_VALUE, min2 = Integer.MAX_VALUE;

        for (int num : nums) {
            if (num > max1) {
                max2 = max1;
                max1 = num;
    } else if (num > max2) {
        max2 = num;
    }

    if (num < min1) {
        min2 = min1;
        min1 = num;
    } else if (num < min2) {
        min2 = num;
    }
}

int product1 = max1 * max2;
int product2 = min1 * min2;

if (product1 > product2) {
    System.out.println("Pair is (" + max1 + ", " + max2 + "), Maximum Product: " + product1);
} else {
    System.out.println("Pair is (" + min1 + ", " + min2 + "), Maximum Product: " + product2);
}
```

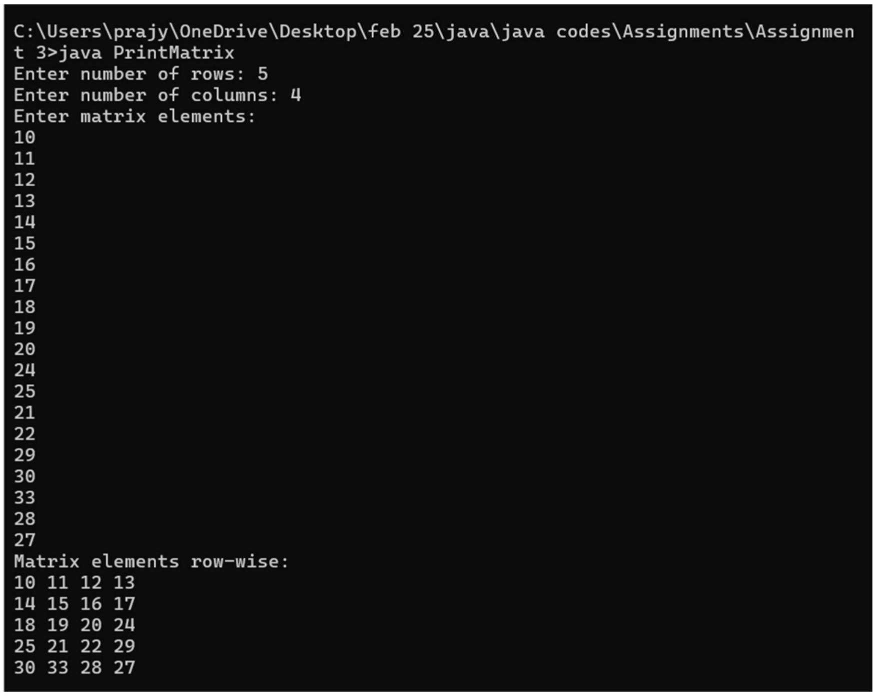
```
    }



    public static void main(String[] args) {

        int[] nums = {2, 3, 5, 7, -7, 5, 8, -5};

        findMaxProduct(nums);

    }

}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>javac MaxProductPair.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>java MaxProductPair
Pair is (8, 7), Maximum Product: 56
```

18. Print a Matrix ○ Given an m x n matrix, print all its elements row-wise.

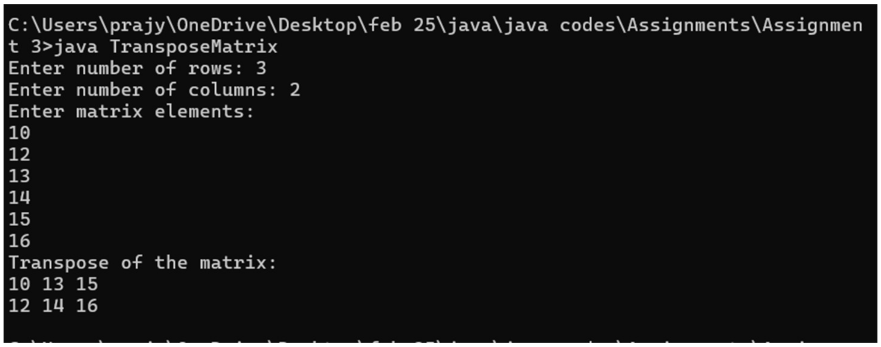
```
import java.util.Scanner;



public class PrintMatrix {

    public static void printMatrix(int[][] matrix) {

        int rows = matrix.length;

        int cols = matrix[0].length;



        System.out.println("Matrix elements row-wise:");

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int m = scanner.nextInt();
        System.out.print("Enter number of columns: ");
        int n = scanner.nextInt();

        int[][] matrix = new int[m][n];
        System.out.println("Enter matrix elements:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }



        printMatrix(matrix);

    }

}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>java PrintMatrix
Enter number of rows: 5
Enter number of columns: 4
Enter matrix elements:
10
11
12
13
14
15
16
17
18
19
20
24
25
21
22
29
30
33
28
27
Matrix elements row-wise:
10 11 12 13
14 15 16 17
18 19 20 24
25 21 22 29
30 33 28 27
```

19. Transpose of a Matrix ○ Given a matrix, return its transpose (swap rows and columns)

```
import java.util.Scanner;



public class TransposeMatrix {

    public static void transposeMatrix(int[][] matrix, int m, int n) {

        int[][] transpose = new int[n][m];
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                transpose[j][i] = matrix[i][j];
            }
        }

        System.out.println("Transpose of the matrix:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                System.out.print(transpose[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int m = scanner.nextInt();
        System.out.print("Enter number of columns: ");
        int n = scanner.nextInt();

        int[][] matrix = new int[m][n];
        System.out.println("Enter matrix elements:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        transposeMatrix(matrix, m, n);

    }
}
```



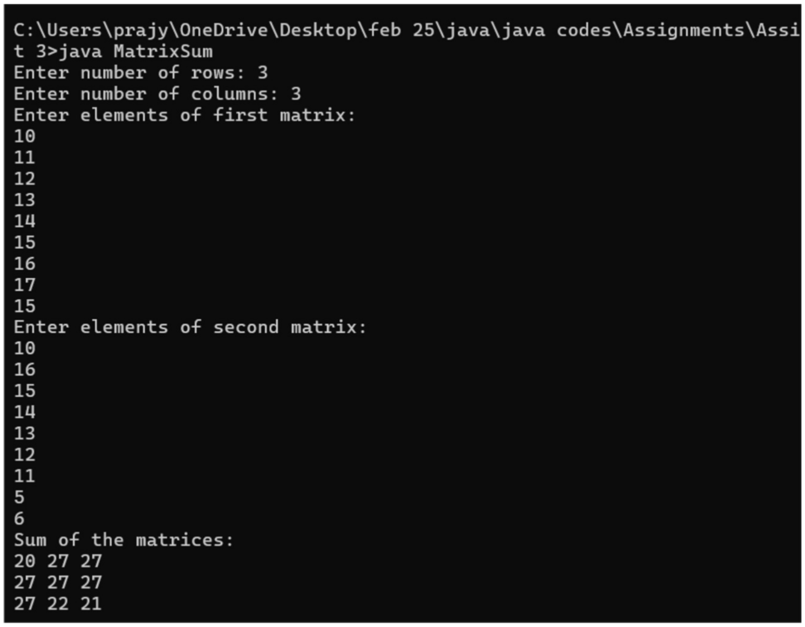
```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>java TransposeMatrix
Enter number of rows: 3
Enter number of columns: 2
Enter matrix elements:
10
12
13
14
15
16
Transpose of the matrix:
10 13 15
12 14 16
```

20. Sum of Two Matrices ○ Given two matrices of the same size, compute their sum.

```java
import java.util.Scanner;


public class MatrixSum {
    public static void sumMatrices(int[][] matrix1, int[][] matrix2, int m, int n) {
        int[][] sum = new int[m][n];


        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                sum[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }


        System.out.println("Sum of the matrices:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(sum[i][j] + " ");
            }
            System.out.println();
        }
```

```
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter number of rows: ");
    int m = scanner.nextInt();
    System.out.print("Enter number of columns: ");
    int n = scanner.nextInt();

    int[][] matrix1 = new int[m][n];
    int[][] matrix2 = new int[m][n];

    System.out.println("Enter elements of first matrix:");
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            matrix1[i][j] = scanner.nextInt();
        }
    }

    System.out.println("Enter elements of second matrix:");
```

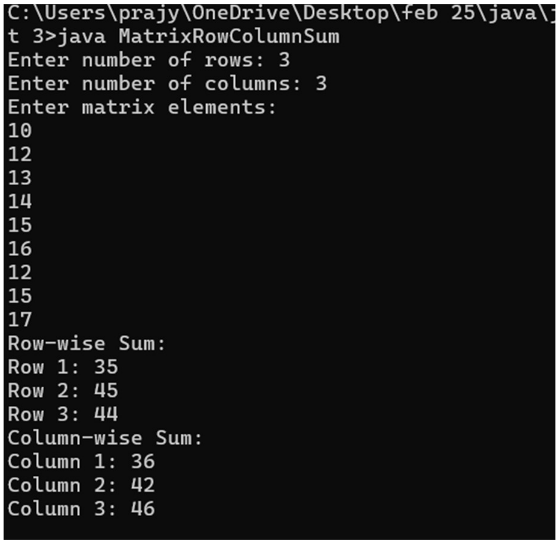
```
        for (int i = 0; i < m; i++) {

            for (int j = 0; j < n; j++) {

                matrix2[i][j] = scanner.nextInt();

            }

        }



        sumMatrices(matrix1, matrix2, m, n);

        scanner.close();

    }

}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assi
t 3>java MatrixSum
Enter number of rows: 3
Enter number of columns: 3
Enter elements of first matrix:
10
11
12
13
14
15
16
17
15
Enter elements of second matrix:
10
16
15
14
13
12
11
5
6
Sum of the matrices:
20 27 27
27 27 27
27 22 21
```

21. Row-wise and Column-wise Sum ○ Find the sum of each row and each column of a given matrix.

```
import java.util.Scanner;

public class MatrixRowColumnSum {
    public static void rowColumnSum(int[][] matrix, int m, int n) {
        System.out.println("Row-wise Sum:");
        for (int i = 0; i < m; i++) {
            int rowSum = 0;
            for (int j = 0; j < n; j++) {
                rowSum += matrix[i][j];
            }
            System.out.println("Row " + (i + 1) + ": " + rowSum);
        }

        System.out.println("Column-wise Sum:");
        for (int j = 0; j < n; j++) {
            int colSum = 0;
            for (int i = 0; i < m; i++) {
                colSum += matrix[i][j];
            }
            System.out.println("Column " + (j + 1) + ": " + colSum);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int m = scanner.nextInt();
        System.out.print("Enter number of columns: ");
        int n = scanner.nextInt();

        int[][] matrix = new int[m][n];
        System.out.println("Enter matrix elements:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        rowColumnSum(matrix, m, n);
    }

}
```

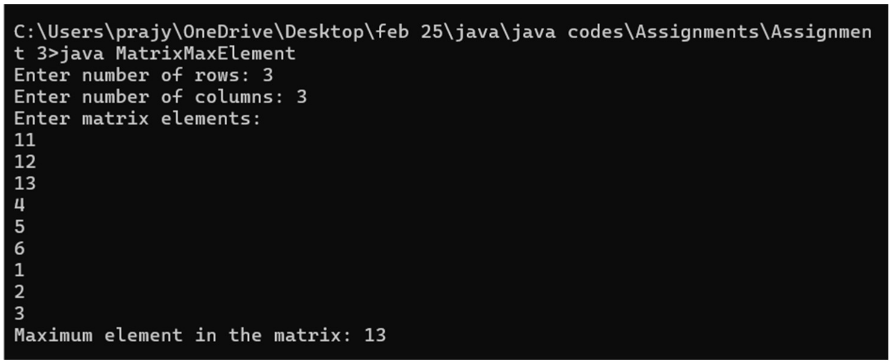


```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\
t 3>java MatrixRowColumnSum
Enter number of rows: 3
Enter number of columns: 3
Enter matrix elements:
10
12
13
14
15
16
12
15
17
Row-wise Sum:
Row 1: 35
Row 2: 45
Row 3: 44
Column-wise Sum:
Column 1: 36
Column 2: 42
Column 3: 46
```

22. Find the Maximum Element in a Matrix ○ Find the largest element in a given matrix.

```
import java.util.Scanner;



public class MatrixMaxElement {

    public static void findMaxElement(int[][] matrix, int m, int n) {

        int maxElement = Integer.MIN_VALUE;



        for (int i = 0; i < m; i++) {

            for (int j = 0; j < n; j++) {

                if (matrix[i][j] > maxElement) {
```

```
                maxElement = matrix[i][j];
            }
        }
    }

    System.out.println("Maximum element in the matrix: " + maxElement);
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter number of rows: ");
    int m = scanner.nextInt();
    System.out.print("Enter number of columns: ");
    int n = scanner.nextInt();

    int[][] matrix = new int[m][n];
    System.out.println("Enter matrix elements:");
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            matrix[i][j] = scanner.nextInt();
            }

        }



        findMaxElement(matrix, m, n);



    }

}
```



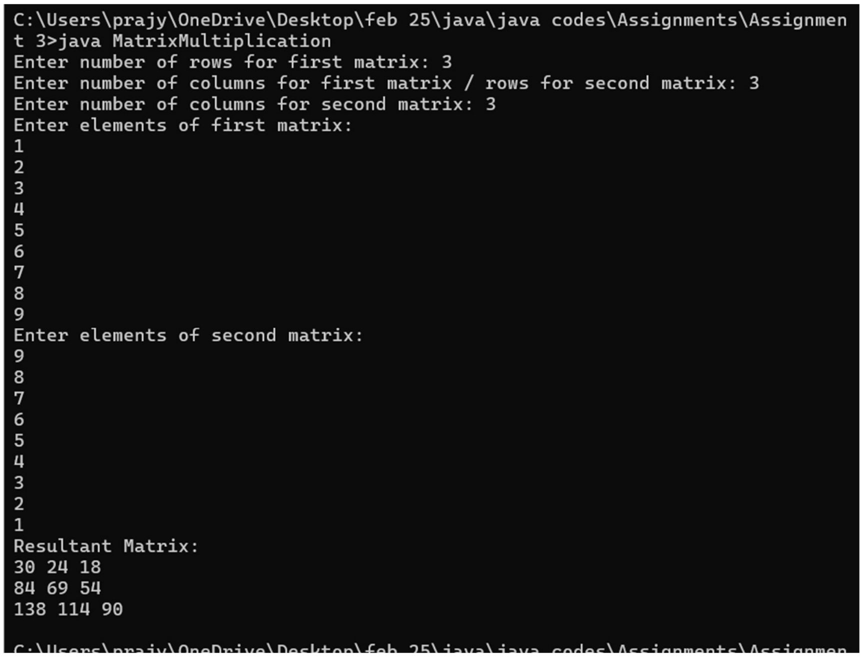
```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>java MatrixMaxElement
Enter number of rows: 3
Enter number of columns: 3
Enter matrix elements:
11
12
13
4
5
6
1
2
3
Maximum element in the matrix: 13
```

23. Matrix Multiplication ○ Multiply two matrices and return the resultant matrix.

```
import java.util.Scanner;



public class MatrixMultiplication {

    public static void multiplyMatrices(int[][] matrix1, int[][] matrix2, int m, int n, int p) {

        int[][] result = new int[m][p];
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < p; j++) {
                for (int k = 0; k < n; k++) {
                    result[i][j] += matrix1[i][k] * matrix2[k][j];
                }
            }
        }

        System.out.println("Resultant Matrix:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < p; j++) {
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of rows for first matrix: ");
```

```
int m = scanner.nextInt();
System.out.print("Enter number of columns for first matrix / rows for second matrix: ");
int n = scanner.nextInt();
System.out.print("Enter number of columns for second matrix: ");
int p = scanner.nextInt();

int[][] matrix1 = new int[m][n];
int[][] matrix2 = new int[n][p];

System.out.println("Enter elements of first matrix:");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        matrix1[i][j] = scanner.nextInt();
    }
}

System.out.println("Enter elements of second matrix:");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < p; j++) {
        matrix2[i][j] = scanner.nextInt();
```

```
            }

        }



        multiplyMatrices(matrix1, matrix2, m, n, p);



    }

}
```



```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 3>java MatrixMultiplication
Enter number of rows for first matrix: 3
Enter number of columns for first matrix / rows for second matrix: 3
Enter number of columns for second matrix: 3
Enter elements of first matrix:
1
2
3
4
5
6
7
8
9
Enter elements of second matrix:
9
8
7
6
5
4
3
2
1
Resultant Matrix:
30 24 18
84 69 54
138 114 90

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
```