JAVA LAB WEEK 3 PROGRAMS

ONE DIMENSIONAL ARRAY

1. Write a program to calculate the sum of all elements in an integer array

```
import java.util.Scanner;
public class SumOfElements {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        int sum = 0;
        for (int i = 0; i < n; i++) {
            sum += arr[i];
        }
        System.out.println("Sum of all elements: " + sum);
    }
}
```

OUTPUT

```
Enter the size of the array: 5
Enter the elements of the array:
1 2 3 4 5
Sum of all elements: 15
```

2.Write a program to find and print the maximum and minimum element in an integer array

```
import java.util.Scanner;


public class MaxMinArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        int max = arr[0], min = arr[0];
        for (int i = 1; i < n; i++) {
            if (arr[i] > max) max = arr[i];
            if (arr[i] < min) min = arr[i];
        }
        System.out.println("Maximum: " + max);
        System.out.println("Minimum: " + min);
    }
}
```

OUTPUT

```
Enter the size of the array: 5
Enter the elements of the array:
7 8 6 5 4
Maximum: 8
Minimum: 4
```

3. Write a program that removes duplicate elements from an array and returns a new array.

```
public class duplicate{
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the size of the array: ");
      int n = sc.nextInt();
      int[] arr = new int[n];
      System.out.println("Enter the elements of the array: ");
      for (int i = 0; i < n; i++) {
         arr[i] = sc.nextInt();
      }
      Arrays.sort(arr);
      System.out.print(arr[0]+" ");
      for (int i = 1; i < n; i++) {
         if (arr[i]!=arr[i-1])System.out.print(arr[i]+" ");
      }
   }
}
```

OUTPUT

```
Enter the size of the array: 5
Enter the elements of the array:
7 8 6 7 8
6 7 8
```

4.Write a program that rotates an array to the right by a specified number of positions.

Sample input and output

Enter the size of the array: 5

Enter the elements of the array: 1 2 3 4 5

Enter the number of positions to rotate: 2

Array after rotation: [4, 5, 1, 2, 3]

```
import java.util.Scanner;

public class RotateArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter the number of positions to rotate: ");
        int k = sc.nextInt();
        k = k % n;
        rotateArray(arr, n, k);
        System.out.print("Array after rotation: ");
        for (int i : arr) {
            System.out.print(i + " ");
        }
    }
    private static void rotateArray(int[] arr, int n, int k) {
        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--;
        }
    }
}
```

OUTPUT

```
Enter the size of the array: 5
Enter the elements of the array:
7 8 9 4 5
Enter the number of positions to rotate: 2
Array after rotation: 4 5 7 8 9
```

5.Write a program that merges two sorted integer arrays into a single sorted array.

Sample input and output
Enter the size of the first array: 3
Enter the elements of the first sorted array: 1 3 5
Enter the size of the second array: 4
Enter the elements of the second sorted array: 2 4 6 8
Merged sorted array: [1, 2, 3, 4, 5, 6, 8]

```
import java.util.Scanner;

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

```
Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the first array: ");
int n1 = sc.nextInt();
int[] arr1 = new int[n1];
System.out.println("Enter the elements of the first sorted array: ");
for (int i = 0; i < n1; i++) {
    arr1[i] = sc.nextInt();
}

System.out.print("Enter the size of the second array: ");
int n2 = sc.nextInt();
int[] arr2 = new int[n2];
System.out.println("Enter the elements of the second sorted array: ");
for (int i = 0; i < n2; i++) {
    arr2[i] = sc.nextInt();
}

int[] merged = new int[n1 + n2];
int i = 0, j = 0, k = 0;

while (i < n1 && j < n2) {
    if (arr1[i] < arr2[j]) {
        merged[k++] = arr1[i++];
    } else {
        merged[k++] = arr2[j++];
    }
}
while (i < n1) {
    merged[k++] = arr1[i++];
```

```
        }
        while (j < n2) {
            merged[k++] = arr2[j++];
        }
        System.out.print("Merged sorted array: [");
        for (int l = 0; l < merged.length; l++) {
            System.out.print(merged[l]);
            if (l < merged.length - 1) {
                System.out.print(", ");
            }
        }
        System.out.println("]");
    }
}
```

OUTPUT

```
Enter the size of the first array: 5
Enter the elements of the first sorted array:
1 2 3 4 5
Enter the size of the second array: 3
Enter the elements of the second sorted array:
7 8 9
Merged sorted array: [1, 2, 3, 4, 5, 7, 8, 9]
```

6.Write a program that finds all unique triplets in an array that sum up to zero.

Sample input and output

Enter the size of the array: 6

Enter the elements of the array: -1 0 1 2 -1 -4

Triplet found: [-1, -1, 2] Triplet found: [-1, 0, 1]

```
import java.util.Scanner;
public class TripletSumZero {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (arr[i] > arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        for (int i = 0; i < n - 2; i++) {
            if (i == 0 || (i > 0 && arr[i] != arr[i - 1])) {
                int left = i + 1, right = n - 1;
                while (left < right) {
                    int sum = arr[i] + arr[left] + arr[right];
                    if (sum == 0) {
                  System.out.println("Triplet found: [" + arr[i] + ", " + arr[left] + ", " + arr[right] + "]");
                        while (left < right && arr[left] == arr[left + 1]) left++;
                        while (left < right && arr[right] == arr[right - 1]) right--;
```

```
                left++;
                right--;
            } else if (sum < 0) {
                left++;
            } else {
                right--;
            }
        }
    }
  }
}
```

OUTPUT

```
Enter the size of the array: 6
Enter the elements of the array:
-1 0 1 2 -1 -4
Triplet found: [-1, -1, 2]
Triplet found: [-1, 0, 1]
```

TWO DIMENSIONAL ARRAY

1.Write a program that computes the transpose of a given 2D matrix.

```
import java.util.Scanner;
public class MatrixTranspose {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter the number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter the number of columns: ");
        int cols = sc.nextInt();

        int[][] matrix = new int[rows][cols];
        System.out.println("Enter the elements of the matrix: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }

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

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

OUTPUT

```
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the matrix:
 1 2 3 4 5 6 7 8 91 2 3 4 5 6 7 8 9
Transpose of the matrix:
1 4 7
2 5 8
3 6 9
```

2.Write a program that prints the elements of a 2D array in spiral order.

Sample input and output
Enter the number of rows: 3
Enter the number of columns: 4
Enter the elements of the matrix:
1 2 3 4
5 6 7 8
9 10 11 12
Spiral order traversal: 1 2 3 4 8 12 11 10 9 5 6 7

```
import java.util.Scanner;
public class SpiralOrderMatrix {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter the number of columns: ");
        int cols = sc.nextInt();

        int[][] matrix = new int[rows][cols];
        System.out.println("Enter the elements of the matrix: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
```

```
            matrix[i][j] = sc.nextInt();
        }
    }

    System.out.print("Spiral order traversal: ");
    int top = 0, bottom = rows - 1, left = 0, right = cols - 1;

    while (top <= bottom && left <= right) {
        for (int i = left; i <= right; i++) {
            System.out.print(matrix[top][i] + " ");
        }
        top++;
        for (int i = top; i <= bottom; i++) {
            System.out.print(matrix[i][right] + " ");
        }
        right--;
        if (top <= bottom) {
            for (int i = right; i >= left; i--) {
                System.out.print(matrix[bottom][i] + " ");
            }
            bottom--;
        }
        if (left <= right) {
            for (int i = bottom; i >= top; i--) {
                System.out.print(matrix[i][left] + " ");
            }
            left++;
        }
    }
}   }
```

OUTPUT

```
Enter the number of rows: 3
Enter the number of columns: 4
Enter the elements of the matrix:
1 2 3 4 5 6 7 8 9 10 11 12
Spiral order traversal: 1 2 3 4 8 12 11 10 9 5 6 7
```

3.Write a program that sets the entire row and column to zero if an element is zero in a given 2D array.

Sample input and output
Enter the number of rows: 3
Enter the number of columns: 4
Enter the elements of the matrix:
1 2 3 4
5 0 7 8
9 10 11 12
Matrix after setting zeroes:
1 0 3 4
0 0 0 0
9 0 11 12

```
import java.util.Scanner;
public class SetMatrixZeroes {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter the number of columns: ");
        int cols = sc.nextInt();

        int[][] matrix = new int[rows][cols];
        System.out.println("Enter the elements of the matrix: ");
```

```
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        matrix[i][j] = sc.nextInt();
    }
}

boolean[] rowZero = new boolean[rows];
boolean[] colZero = new boolean[cols];
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        if (matrix[i][j] == 0) {
            rowZero[i] = true;
            colZero[j] = true;
        }
    }
}
for (int i = 0; i < rows; i++) {
    if (rowZero[i]) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = 0;
        }
    }
}
for (int j = 0; j < cols; j++) {
    if (colZero[j]) {
        for (int i = 0; i < rows; i++) {
            matrix[i][j] = 0;
        }
    }
}
```

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

OUTPUT

```
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the matrix:
1 5 6 4 0 3 0 1 8
Matrix after setting zeroes:
0 0 6
0 0 0
0 0 0
```

4.Write a program to print the below series

```
public class InnerReducingPattern {
    public static void main(String[] args) {
        int n = 7;
        int[][] matrix = new int[n][n];

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                int minDist = Math.min(Math.min(i, j), Math.min(n - 1 - i, n - 1 - j));
                matrix[i][j] = n - minDist;
            }
        }
```

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

OUTPUT

```
7 7 7 7 7 7 7
7 6 6 6 6 6 7
7 6 5 5 5 6 7
7 6 5 4 5 6 7
7 6 5 5 5 6 7
7 6 6 6 6 6 7
7 7 7 7 7 7 7
```