

2D - ARRAY

1. Creation of 2D Array

```
public class creationof2D {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int matrix [][] = new int [rows][cols];
        // we can also find the length by: rows = matrix.length & cols = matrix[0].length
        System.out.print("Enter Elements: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix [i][j] = sc.nextInt();
            }
        }
        System.out.println("Matrix of "+rows+" * "+cols+" is:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

2. Count specific no. in an Array

```
public class Ques1 {
    public static void count_num (int matrix[][], int num) {
        int row = matrix.length, col = matrix[0].length;
        int count = 0;
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                if (matrix[i][j] == num) {
                    count++;
                }
            }
        }
        System.out.println("No. of times " +num+ " appears in 2D Array is: " +count);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter a Number: ");
        int num = sc.nextInt();
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int arr[][] = new int[rows][cols];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                arr[i][j] = sc.nextInt();
            }
        }
        count_num(arr, num);
    }
}
```

3. Searching in 2D Array

```
import java.util.Scanner;

public class searchingin2D {

    public static boolean search (int matrix[][], int key) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                if (matrix[i][j] == key) {
                    System.out.print("Found at index: (" + (i+1) + ", " + (j+1) + ")");
                    return true;
                }
            }
        }
        System.out.println("NOT FOUND !!!");
        return false;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int matrix [][] = new int [rows][cols];
        // we can also find the length by: rows = matrix.length & cols = matrix[0].length
        System.out.print("Enter Elements: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix [i][j] = sc.nextInt();
            }
        }
        System.out.println("Matrix of "+rows+" * "+cols+" is:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
        search(matrix, 5);
    }
}
```

4. Diagonal Sum

```
import java.util.Scanner;

public class diagonalSum {

    public static int diagonal_sum(int matrix[][]) {
        int sum = 0;
        /* TIME COMPLEXITY: O(n^2)
        for (int i=0; i<matrix.length; i++) {
            for (int j=0; j<matrix[0].length; j++) {
                if (i == j) {
                    sum += matrix[i][j];
                }
                else if (i+j == matrix.length-1) {
                    sum += matrix[i][j];
                }
            }
        }
        */
        // TIME COMPLEXITY: O(n)
        for (int i=0; i<matrix.length; i++) {
            sum += matrix[i][i];
            if (i != matrix.length-1-i) {
                sum += matrix[i][matrix.length-i-1];
            }
        }
        return sum;
    }

    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int row = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int matrix [][] = new int [row][cols];
        System.out.println("Enter Elements: ");
        for (int i=0; i<row; i++) {
            for (int j=0; j<cols; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        System.out.print("Diagonal Sum is: " + diagonal_sum(matrix));
    }
}
```

5. Searching in Sorted Matrix

```
import java.util.Scanner;

public class searchinSortedMatrix1 {

    public static boolean staircase_search(int matrix[][], int key) {
        int row = 0, col = matrix[0].length-1;
        while (row < matrix.length && col >= 0) {
            if (matrix[row][col] == key) {
                System.out.print("Found Key at (" + row + ", " + col + ")");
                return true;
            }
            else if (key < matrix[row][col]) {
                col--;
            }
            else {
                row++;
            }
        }
        System.out.println("Key no Found !!!");
        return false;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter key to be search: ");
        int key = sc.nextInt();
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int matrix[][] = new int [rows][cols];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        staircase_search(matrix, key);
    }
}
```

6. Spiral Matrix

```
import java.util.Scanner;

public class spiralMatrix {

    public static void spiral_array(int matrix[][]) {
        int startRow = 0;
        int startCol = 0;
        int endRow = matrix.length-1;
        int endCol = matrix[0].length-1;
        while (startRow <= endRow && startCol <= endCol) {
            // TOP
            for (int j = startCol; j <= endCol; j++) {
                System.out.print(matrix[startRow][j] + " ");
            }
            // RIGHT
            for (int i = startRow+1; i <= endRow; i++) {
                System.out.print(matrix[i][endCol] + " ");
            }
            // BOTTOM
            for (int j = endCol-1; j >= startCol; j--) {
                if (startRow == endRow) {
                    break;
                }
                System.out.print(matrix[endRow][j] + " ");
            }
            // LEFT
            for (int i = endRow-1; i >= startRow+1; i--) {
                if (startCol == endCol) {
                    break;
                }
                System.out.print(matrix[i][startCol] + " ");
            }
            // UPDATING ROWS & COLUMNS
            startRow++;
            startCol++;
            endRow--;
            endCol--;
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int row = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int matrix[][] = new int [row][cols];
        System.out.println("Enter elements for " +row+ " * " +cols+ " matrix: ");
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < cols; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        System.out.print("Printing... Spiral Matrix: ");
        spiral_array(matrix);
    }
}
```

7. Sum of specific row

```
import java.util.Scanner;

public class Ques2 {

    public static void sum_of_row (int matrix[][], int row_no) {
        int sum = 0;
        for (int i = 0; i < matrix[row_no].length; i++) {
            sum += matrix[row_no][i];
        }
        System.out.print("Sum of " + row_no + " row is: " + sum);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the Row no. : ");
        int row_no = sc.nextInt();
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int arr[][] = new int[rows][cols];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                arr[i][j] = sc.nextInt();
            }
        }
        sum_of_row(arr, row_no);
    }
}
```

8. Transpose of an Array

```
import java.util.Scanner;

public class Ques3 {

    public static void print_matrix (int matrix[][]) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                System.out.print(matrix[i][j]+ " ");
            }
            System.out.println();
        }
    }

    public static void transpose (int matrix[][]) {
        int row = matrix.length, cols = matrix[0].length;
        int transpose_matrix[][] = new int [row][cols];
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < cols; j++) {
                transpose_matrix[j][i] = matrix[i][j];
            }
        }
        System.out.println("Transpose Matrix is: ");
        print_matrix(transpose_matrix);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Columns: ");
        int cols = sc.nextInt();
        int arr[][] = new int[rows][cols];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                arr[i][j] = sc.nextInt();
            }
        }
        transpose(arr);
    }
}
```