

Assignment - 2D arrays problem solving in java

2D Array Programs

1. Read and Print a 2D Array

Input:

1 2 3

4 5 6

Output:

1 2 3

4 5 6

Program :-

Import java.util.Scanner;

```
public class ReadandPrint {  
    static void readAndPrintMatrix() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter rows: ");  
        int rows = sc.nextInt();  
        System.out.print("Enter cols: ");  
        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();  
        System.out.println("Matrix:");  
        for (int i = 0; i < rows; i++) {  
            for (int j = 0; j < cols; j++)  
                System.out.print(arr[i][j] + " ");  
            System.out.println();  
        }  
    }  
}
```

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

2. Find the Sum of All Elements

Input: 1 2

3 4

Output: 10

Program :-

Import java.util.Scanner;

```
public class Sumofele {  
    static void sumOfElements() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter rows: ");  
        int rows = sc.nextInt();  
        System.out.print("Enter cols: ");  
        int cols = sc.nextInt();  
        int[][] arr = new int[rows][cols];  
        int sum = 0;  
        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 += arr[i][j];  
            }  
        System.out.println("Sum = " + sum);  
    }  
  
    public static void main(String[] args) {  
        sumOfElements();  
    }  
}
```

3. Row-wise Sum of Elements

Input: 1 2 3

4 5 6

Output:

Row 1 Sum = 6

Row 2 Sum = 15

Program :-

```
Import java.util.Scanner;
```

```
public class RowSum {  
    static void rowWiseSum() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter rows: ");  
        int rows = sc.nextInt();  
        System.out.print("Enter cols: ");  
        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();  
        for (int i = 0; i < rows; i++) {  
            int sum = 0;  
            for (int j = 0; j < cols; j++)  
                sum += arr[i][j];  
            System.out.println("Row " + (i + 1) + " Sum = " + sum);  
        }  
    }  
  
    public static void main(String[] args) {  
        rowWiseSum();  
    }  
}
```

4. Column-wise Sum of Elements

Input: 1 2

3 4

5 6

Output:

Col 1 Sum = 9

Col 2 Sum = 12

Program :-

```
import java.util.Scanner;
```

```
public class Colsum {  
    static void colWiseSum() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter rows: ");  
        int rows = sc.nextInt();  
        System.out.print("Enter cols: ");  
        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();  
        for (int j = 0; j < cols; j++) {  
            int sum = 0;  
            for (int i = 0; i < rows; i++)  
                sum += arr[i][j];  
            System.out.println("Col " + (j + 1) + " Sum = " + sum);  
        }  
    }  
  
    public static void main(String[] args) {  
        colWiseSum();  
    }  
}
```

}

5. Find Maximum Element

Input: 1 9

3 7

Output: Max = 9

Program :-

```
import java.util.Scanner;
```

```
public class Findmax {  
    static void findMax() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter rows: ");  
        int rows = sc.nextInt();  
        System.out.print("Enter cols: ");  
        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();  
        int max = arr[0][0];  
        for (int i = 0; i < rows; i++)  
            for (int j = 0; j < cols; j++)  
                if (arr[i][j] > max)  
                    max = arr[i][j];  
        System.out.println("Max = " + max);  
    }  
  
    public static void main(String[] args) {  
        findMax();  
    }  
}
```

6. Find Minimum Element

Input: 8 2

6 5

Output: Min = 2

Program :-

```
import java.util.Scanner;

public class Findmin {
    static void findMin() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter cols: ");
        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();
        int min = arr[0][0];
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                if (arr[i][j] < min)
                    min = arr[i][j];
        System.out.println("Min = " + min);
    }

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

7. Search an Element in Matrix

Input Matrix:

1 2 3

4 5 6

Search: 5

Output: Found at (1,1)

Program :-

```
import java.util.Scanner;
```

```
public class Searchele {
    static void searchElement() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter cols: ");
        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();
        System.out.print("Enter element to search: ");
        int key = sc.nextInt();
        boolean found = false;
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (arr[i][j] == key) {
                    System.out.println("Found at (" + i + ", " +
j + ")");
                    found = true;
                    break;
                }
            }
        }
        if (!found)
            System.out.println("Not Found");
    }
}
```

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

8. Count Even and Odd Numbers

Input: 1 2 3

4 5 6

Output:

Even = 3, Odd = 3

Program :-

```
import java.util.Scanner;
```

```
public class CountEvOd {
    static void countEvenOdd() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter cols: ");
        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();
        int even = 0, odd = 0;
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++) {
                if (arr[i][j] % 2 == 0)
                    even++;
                else
                    odd++;
            }
    }
}
```



```

        System.out.println("Even = " + even + ", Odd = " + odd);
    }

    public static void main(String[] args) {
        countEvenOdd();
    }
}

```

9. Calculate Average of All Elements

Input: 2 4

6 8

Output: Average = 5.0

Program :-

```

import java.util.Scanner;

public class AvgMat {
    static void averageMatrix() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter cols: ");
        int cols = sc.nextInt();
        int[][] arr = new int[rows][cols];
        System.out.println("Enter elements:");
        int sum = 0;
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++) {
                arr[i][j] = sc.nextInt();
                sum += arr[i][j];
            }
        double avg = (double) sum / (rows * cols);
        System.out.println("Average = " + avg);
    }
}

```

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

10. Print Left Diagonal Elements

Input:

1 2 3

4 5 6

7 8 9

Output: 1 5 9

Program :-

```
import java.util.Scanner;  
  
public class DiagLeft {  
    static void leftDiagonal() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter size: ");  
        int n = sc.nextInt();  
        int[][] arr = new int[n][n];  
        System.out.println("Enter elements:");  
        for (int i = 0; i < n; i++)  
            for (int j = 0; j < n; j++)  
                arr[i][j] = sc.nextInt();  
        System.out.print("Left Diagonal: ");  
        for (int i = 0; i < n; i++)  
            System.out.print(arr[i][i] + " ");  
        System.out.println();  
    }  
  
    public static void main(String[] args) {  
        leftDiagonal();  
    }  
}
```

}

11.Print Right Diagonal Elements

Input:

1 2 3

4 5 6

7 8 9

Output: 3 5 7

Program :-

```
import java.util.Scanner;
```

```
public class Diagright {  
    static void rightDiagonal() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter size: ");  
        int n = sc.nextInt();  
        int[][] arr = new int[n][n];  
        System.out.println("Enter elements:");  
        for (int i = 0; i < n; i++)  
            for (int j = 0; j < n; j++)  
                arr[i][j] = sc.nextInt();  
        System.out.print("Right Diagonal: ");  
        for (int i = 0; i < n; i++)  
            System.out.print(arr[i][n - 1 - i] + " ");  
        System.out.println();  
    }  
  
    public static void main(String[] args) {  
        rightDiagonal();  
    }  
}
```

12.Print All Boundary Elements

Input:

1 2 3

4 5 6

7 8 9

Output: 1 2 3 6 9 8 7 4

Program :-

```
import java.util.Scanner;
```

```
public class Boundele {  
    static void boundaryElements() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter rows: ");  
        int rows = sc.nextInt();  
        System.out.print("Enter cols: ");  
        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();  
        System.out.print("Boundary Elements: ");  
        for (int j = 0; j < cols; j++)  
            System.out.print(arr[0][j] + " ");  
        for (int i = 1; i < rows; i++)  
            System.out.print(arr[i][cols - 1] + " ");  
        for (int j = cols - 2; j >= 0; j--)  
            if (rows > 1)  
                System.out.print(arr[rows - 1][j] + " ");  
        for (int i = rows - 2; i > 0; i--)  
            if (cols > 1)  
                System.out.print(arr[i][0] + " ");  
        System.out.println();  
    }  
}
```

```

    }

    public static void main(String[] args) {
        boundaryElements();
    }
}

```

13. Replace All Negative Numbers with 0

Input:

1 -2 3

-4 5 -6

Output:

1 0 3

0 5 0

Program :-

```

import java.util.Scanner;

public class Replaceneg {
    static void replaceNegatives() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter cols: ");
        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();
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (arr[i][j] < 0)

```

```

        arr[i][j] = 0;
        System.out.print(arr[i][j] + " ");
    }
    System.out.println();
}
}

public static void main(String[] args) {
    replaceNegatives();
}
}

```

14. Print Row with Maximum Sum

Input:

1 2 3

9 8 7

Output: Row 2 has maximum sum = 24

Program :-

```

import java.util.Scanner;

public class RowmaxSum {
    static void rowMaxSum() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter cols: ");
        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();
    }
}

```

```

        int maxSum = Integer.MIN_VALUE, rowIndex = 0;
        for (int i = 0; i < rows; i++) {
            int sum = 0;
            for (int j = 0; j < cols; j++)
                sum += arr[i][j];
            if (sum > maxSum) {
                maxSum = sum;
                rowIndex = i;
            }
        }
        System.out.println("Row " + (rowIndex + 1) + " has maximum
sum = " + maxSum);
    }

    public static void main(String[] args) {
        rowMaxSum();
    }
}

```

15. Print Column with Maximum Sum

Input:

1 4

2 5

3 6

Output: Column 2 has maximum sum = 15

Program :-

```

import java.util.Scanner;

public class ColmaxSum {
    static void colMaxSum() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows: ");
    }
}

```

```

int rows = sc.nextInt();
System.out.print("Enter cols: ");
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();
int maxSum = Integer.MIN_VALUE, colIndex = 0;
for (int j = 0; j < cols; j++) {
    int sum = 0;
    for (int i = 0; i < rows; i++)
        sum += arr[i][j];
    if (sum > maxSum) {
        maxSum = sum;
        colIndex = j;
    }
}
System.out.println("Column " + (colIndex + 1) + " has maximum
sum = " + maxSum);
}

public static void main(String[] args) {
    colMaxSum();
}
}

```

Intermediate 2D Array Programs

1. Transpose of Matrix

Input:

1 2 3

4 5 6

Output:

1 4
2 5
3 6

Program :-

```
import java.util.Scanner;

public class Transpose {

    public static void transposeMatrix(int[][] matrix, int rows, int cols) {
        System.out.println("Transpose of Matrix:");
        for (int i = 0; i < cols; i++) {
            for (int j = 0; j < rows; j++) {
                System.out.print(matrix[j][i] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int rows = 2, cols = 3;
        int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 } };
        transposeMatrix(matrix, rows, cols);
    }
}
```

2. Check if Matrix is Symmetric

Input:

1 2
2 1

Output: Symmetric

Program :-

```
import java.util.Scanner;

public class Symmetric {

    public static void checkSymmetric(int[][] matrix, int n) {
        boolean flag = true;

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (matrix[i][j] != matrix[j][i]) {
                    flag = false;
                    break;
                }
            }
        }

        if (flag) {
            System.out.println("Symmetric Matrix");
        } else {
            System.out.println("Not Symmetric Matrix");
        }
    }

    public static void main(String[] args) {
        int[][] matrix = { { 1, 2 }, { 2, 1 } };

        checkSymmetric(matrix, 2);
    }
}
```

3. Check if Identity Matrix

Input:

1 0 0

0 1 0

0 0 1

Output: Identity Matrix

Program :-

```
public class Identity {  
  
    public static void checkIdentity(int[][] matrix, int n) {  
        boolean flag = true;  
  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < n; j++) {  
                if ((i == j && matrix[i][j] != 1) || (i != j &&  
matrix[i][j] != 0)) {  
                    flag = false;  
                    break;  
                }  
            }  
        }  
  
        if (flag) {  
            System.out.println("Identity Matrix");  
        } else {  
            System.out.println("Not Identity Matrix");  
        }  
    }  
  
    public static void main(String[] args) {  
        int[][] matrix = { { 1, 0, 0 }, { 0, 1, 0 }, { 0, 0, 1 } };  
  
        checkIdentity(matrix, 3);  
    }  
}
```

4. Matrix Addition

A:

1 2

3 4

B:

5 6

7 8

Output:

6 8

10 12

Program :-

```
public class Matadd {  
  
    public static void matrixAddition(int[][] A, int[][] B, int r, int c) {  
        System.out.println("Matrix Addition:");  
        for (int i = 0; i < r; i++) {  
            for (int j = 0; j < c; j++) {  
                System.out.print((A[i][j] + B[i][j]) + " ");  
            }  
            System.out.println();  
        }  
    }  
  
    public static void main(String[] args) {  
        int[][] A = { { 1, 2 }, { 3, 4 } };  
  
        int[][] B = { { 5, 6 }, { 7, 8 } };  
  
        matrixAddition(A, B, 2, 2);  
    }  
}
```

5. Matrix Subtraction

A:

5 5

5 5

B:

2 3

4 1

Output:

3 2

1 4

Program :-

```
public class Matsub {  
  
    public static void matrixSubtraction(int[][] A, int[][] B, int r, int c) {  
        System.out.println("Matrix Subtraction:");  
        for (int i = 0; i < r; i++) {  
            for (int j = 0; j < c; j++) {  
                System.out.print((A[i][j] - B[i][j]) + " ");  
            }  
            System.out.println();  
        }  
    }  
  
    public static void main(String[] args) {  
        int[][] A = { { 5, 5 }, { 5, 5 } };  
  
        int[][] B = { { 2, 3 }, { 4, 1 } };  
  
        matrixSubtraction(A, B, 2, 2);  
    }  
}
```

6. Matrix Multiplication

A:

1 2

3 4

B:

2 0

1 2

Output:

4 4

10 8

Program :-

```
public class Matmul {  
  
    public static void matrixMultiplication(int[][] A, int[][] B, int r1, int c1, int  
c2) {  
        int[][] C = new int[r1][c2];  
        System.out.println("Matrix Multiplication:");  
  
        for (int i = 0; i < r1; i++) {  
            for (int j = 0; j < c2; j++) {  
                for (int k = 0; k < c1; k++) {  
                    C[i][j] += A[i][k] * B[k][j];  
                }  
                System.out.print(C[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
  
    public static void main(String[] args) {  
        int[][] A = { { 1, 2 }, { 3, 4 } };  
  
        int[][] B = { { 2, 0 }, { 1, 2 } };
```

```
        matrixMultiplication(A, B, 2, 2, 2);
    }
}
```

7. Sum of Upper Triangular Elements

Input:

1 2 3

0 4 5

0 0 6

Output: 21

Program :-

```
public class SumuppTri {

    public static void sumUpperTriangle(int[][] matrix, int n) {
        int sum = 0;

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

        System.out.println("Sum of Upper Triangular Elements = " + sum);
    }

    public static void main(String[] args) {
        int[][] matrix = { { 1, 2, 3 }, { 0, 4, 5 }, { 0, 0, 6 } };

        sumUpperTriangle(matrix, 3);
    }
}
```

8. Sum of Lower Triangular Elements

Input:

1 0 0

2 3 0

4 5 6

Output: 21

Program :-

```
public class SumlowTri {  
  
    public static void sumLowerTriangle(int[][] matrix, int n) {  
        int sum = 0;  
  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j <= i; j++) {  
                sum += matrix[i][j];  
            }  
        }  
  
        System.out.println("Sum of Lower Triangular Elements = " + sum);  
    }  
  
    public static void main(String[] args) {  
        int[][] matrix = { { 1, 0, 0 }, { 2, 3, 0 }, { 4, 5, 6 } };  
  
        sumLowerTriangle(matrix, 3);  
    }  
}
```

9. Check Sparse Matrix

Input:

0 0 1

0 0 0

2 0 0

Output: Sparse Matrix

Program :-

```
public class Sparsemat {  
  
    public static void checkSparse(int[][] matrix, int r, int c) {  
        int zeroCount = 0;  
        int total = r * c;  
  
        for (int i = 0; i < r; i++) {  
            for (int j = 0; j < c; j++) {  
                if (matrix[i][j] == 0) {  
                    zeroCount++;  
                }  
            }  
        }  
  
        if (zeroCount > total / 2) {  
            System.out.println("Sparse Matrix");  
        } else {  
            System.out.println("Not Sparse Matrix");  
        }  
    }  
  
    public static void main(String[] args) {  
        int[][] matrix = { { 0, 0, 1 }, { 0, 0, 0 }, { 2, 0, 0 } };  
  
        checkSparse(matrix, 3, 3);  
    }  
}
```

10. Spiral Order Traversal

Input:

1 2 3

4 5 6

7 8 9

Output: 1 2 3 6 9 8 7 4 5

Program :-

```
public class SpiralOrd {

    public static void spiralOrder(int[][] matrix, int r, int c) {
        System.out.println("Spiral Order Traversal:");
        int top = 0, bottom = r - 1;
        int left = 0, right = c - 1;

        while (top <= bottom && left <= right) {
            // Top row
            for (int i = left; i <= right; i++) {
                System.out.print(matrix[top][i] + " ");
            }
            top++;

            // Right column
            for (int i = top; i <= bottom; i++) {
                System.out.print(matrix[i][right] + " ");
            }
            right--;

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

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

```

        }
        left++;
    }
}
System.out.println();
}

public static void main(String[] args) {
    int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

    spiralOrder(matrix, 3, 3);
}
}

```

11. Snake Pattern Traversal

Input:

```

1 2 3
4 5 6
7 8 9

```

Output: 1 2 3 6 5 4 7 8 9

Program :-

```

public class SnakePat {

    public static void snakePattern(int[][] matrix, int r, int c) {
        System.out.println("Snake Pattern Traversal:");
        for (int i = 0; i < r; i++) {
            if (i % 2 == 0) {
                for (int j = 0; j < c; j++) {
                    System.out.print(matrix[i][j] + " ");
                }
            }
            else {
                for (int j = c - 1; j >= 0; j--) {

```

```

                System.out.print(matrix[i][j] + " ");
            }
        }
    }
    System.out.println();
}

public static void main(String[] args) {
    int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

    snakePattern(matrix, 3, 3);
}
}

```

12. Rotate Matrix 90 Degrees Clockwise

Input:

1 2 3

4 5 6

7 8 9

Output:

7 4 1

8 5 2

9 6 3

Program :-

```

public class Rotate90Mat {

    public static void rotate90(int[][] matrix, int n) {
        System.out.println("Matrix Rotated 90 Degrees Clockwise:");
        for (int i = 0; i < n; i++) {
            for (int j = n - 1; j >= 0; j--) {
                System.out.print(matrix[j][i] + " ");
            }
            System.out.println();
        }
    }
}

```

```

    }
}

public static void main(String[] args) {
    int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

    rotate90(matrix, 3);
}
}

```

13. Find Saddle Point (smallest in row, largest in column)

Input:

```

1 2 3
4 5 6
7 8 9

```

Output: Saddle Point = 7

Program :-

```

public class SaddlePoint {

    public static void saddlePoint(int[][] matrix, int n) {
        for (int i = 0; i < n; i++) {
            int rowMin = matrix[i][0];
            int colIndex = 0;

            for (int j = 1; j < n; j++) {
                if (matrix[i][j] < rowMin) {
                    rowMin = matrix[i][j];
                    colIndex = j;
                }
            }

            boolean flag = true;
            for (int k = 0; k < n; k++) {

```

```

        if (matrix[k][colIndex] > rowMin) {
            flag = false;
            break;
        }
    }

    if (flag) {
        System.out.println("Saddle Point = " + rowMin);
        return;
    }

    System.out.println("No Saddle Point");
}

public static void main(String[] args) {
    int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

    saddlePoint(matrix, 3);
}
}

```

14. Check Magic Square (sum of rows, cols, diagonals equal)

Input:

2 7 6

9 5 1

4 3 8

Output: Magic Square

Program :-

```

public class MagicSqr {

    public static void checkMagicSquare(int[][] matrix, int n) {
        int sum = 0;
        // Sum of first row
        for (int j = 0; j < n; j++) {

```

```

        sum += matrix[0][j];
    }

    boolean flag = true;

    // Check sum of rows
    for (int i = 0; i < n; i++) {
        int rowSum = 0;
        for (int j = 0; j < n; j++) {
            rowSum += matrix[i][j];
        }
        if (rowSum != sum)
            flag = false;
    }

    // Check sum of columns
    for (int j = 0; j < n; j++) {
        int colSum = 0;
        for (int i = 0; i < n; i++) {
            colSum += matrix[i][j];
        }
        if (colSum != sum)
            flag = false;
    }

    // Check diagonals
    int d1 = 0, d2 = 0;
    for (int i = 0; i < n; i++) {
        d1 += matrix[i][i];
        d2 += matrix[i][n - i - 1];
    }

    if (d1 != sum || d2 != sum)
        flag = false;

    System.out.println(flag ? "Magic Square" : "Not Magic Square");
}

```

```

    public static void main(String[] args) {
        int[][] matrix = { { 2, 7, 6 }, { 9, 5, 1 }, { 4, 3, 8 } };

        checkMagicSquare(matrix, 3);
    }
}

```

15. Sum of Both Diagonals

Input:

1 2 3

4 5 6

7 8 9

Output:

Left Diagonal Sum = 15

Right Diagonal Sum = 15

Program :-

```

public class DiagonalsSum {

    public static void sumDiagonals(int[][] matrix, int n) {
        int left = 0, right = 0;

        for (int i = 0; i < n; i++) {
            left += matrix[i][i]; // Left diagonal
            right += matrix[i][n - i - 1]; // Right diagonal
        }

        System.out.println("Left Diagonal Sum = " + left);
        System.out.println("Right Diagonal Sum = " + right);
    }

    public static void main(String[] args) {
        int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
    }
}

```



```
    sumDiagonals(matrix, 3);  
  }  
}
```