# Assignment 1 [13th Sept 2025]

**Problem 1:**

Your job:

● Find the team whose total score is the highest (row with maximum sum).

● Report the team’s index (row number) to the King.

● If two or more teams tie with the same score, the earliest team in the scoreboard

should be declared the winner.

Only then will the King crown the winning team with the Golden Abacus Trophy.

**Input**

● First line contains two integers N (number of teams / rows) and M (number of

players per team / columns).

● Next N lines each contain M integers — the scoreboard matrix.

**Output**

● Print the index (0-based) of the team with the highest total score.

**Solution**

// Problem 1 from 13\_Sept.pdf given by Prof. Pradep Kumar

import java.util.\*;

public class RoyalTournament {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int teams = sc.nextInt();

int players = sc.nextInt();

int[][] score = new int[teams][players];

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

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

score[i][j] = sc.nextInt();

}

}

int maxScore = Integer.MIN\_VALUE;

int winningTeam = -1;

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

int teamScore = 0;

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

teamScore += score[i][j];

}

if (teamScore > maxScore) {

maxScore = teamScore;

winningTeam = i;

}

else if (teamScore == maxScore) {

winningTeam = -1; // Tie

}

}

sc.close();

System.out.println(winningTeam);

}

}

Output:

4 4

1 2 3 4

5 6 7 8

9 10 11 12

22 32 1 0

3

**Problem 2:**

**Input**

● First line contains an integer N → size of the chessboard (matrix).

● Next N lines contain N integers → the matrix elements.

**Output**

● Print two integers:

1. Sum of the main diagonal

2. Sum of the secondary diagonal

**Constraints**

● 1 ≤ N ≤ 100

● -10^5 ≤ Matrix[i][j] ≤ 10^5

**Solution**

// Problem 2 from 13\_Sept.pdf given by Prof. Pradep Kumar

import java.util.Scanner;

public class MagicalStrength {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int row = sc.nextInt();

int column = sc.nextInt();

int[][] matrix = new int[row][column];

int mainDiagonalSum = 0;

int secondaryDiagonalSum = 0;

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

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

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

}

}

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

mainDiagonalSum += matrix[i][i];

secondaryDiagonalSum += matrix[i][column - 1 - i];

}

sc.close();

System.out.print(mainDiagonalSum);

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

}

}

Output:

3

3

1 2 3

4 5 6

7 8 9

**Problem 3:**

The King, being curious, wants you to perform this magical row reversal.

Input

● First line contains two integers N (rows) and M (columns).

● Next N lines contain M integers each → the matrix.

Output

● Print the matrix after reversing each row.

Constraints

● 1 ≤ N, M ≤ 100

● -10^5 ≤ Matrix[i][j] ≤ 10^5

Input

3 3

1 2 3

4 5 6

7 8 9

Output

3 2 1

6 5 4

9 8 7

**Solution:**

import java.util.\*;

// Problem 3 from 13\_Sept.pdf given by Prof. Pradep Kumar

public class ScoreboardReversal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int rows = sc.nextInt();

int cols = sc.nextInt();

int[][] matrix = new int[rows][cols];

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

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

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

}

}

// Reversing each row

for (int i = 0; i < matrix.length; i++) {

int start = 0;

int end = matrix[i].length - 1;

while (start < end) {

int temp = matrix[i][start];

matrix[i][start] = matrix[i][end];

matrix[i][end] = temp;

start++;

end--;

}

}

sc.close();

// Output the modified matrix

System.out.println();

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

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

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

}

System.out.println();

}

}

}

**Output:**

3 3

1 2 3

4 5 6

7 8 9

3 2 1

6 5 4

9 8 7

**Problem 4**

Your task is to:

● Construct the pattern using matrix elements (not just 0/1).

● Replace all non-pattern positions with 0.

● Print the resulting matrix.

● Print the sum of all numbers used in the pattern.

**Input Format**

● First line: A character ch (Z, D, A, X).

● Second line: An integer N (3 ≤ N ≤ 20).

**Output Format**

● The N × N matrix where only the chosen pattern positions keep their values, all

others are 0.

● The sum of all values included in the pattern.

**Constraints**

● 3 ≤ N ≤ 20

● Pattern is guaranteed to be one of {Z, D, A, X}

**Solution:**

// This is a problem 1 from Assignment Quesiton 13\_Sept.pdf(not same)

import java.util.Scanner;

public class DrawnPattern {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

char ch = sc.nextLine().charAt(0);

int N = sc.nextInt();

// Characters are only Z, D, A, X which will be drawn as a pattern in a N\*N matrix

// Also print the sum of all numbers used in the pattern

// replace all non pattern positions with 0

int sum = 0;

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

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

if (ch == 'Z') {

if (i == 0 || i == N - 1 || i + j == N - 1) {

System.out.print((i + 1) + " ");

sum += (i + 1);

} else {

System.out.print("0 ");

}

} else if (ch == 'D') {

if (j == 0 || j == N - 1 || i == j) {

System.out.print((i + 1) + " ");

sum += (i + 1);

} else {

System.out.print("0 ");

}

} else if (ch == 'A') {

if (i == 0 || i == N - 1 || i == j || i + j == N - 1) {

System.out.print((i + 1) + " ");

sum += (i + 1);

} else {

System.out.print("0 ");

}

} else if (ch == 'X') {

if (i == j || i + j == N - 1) {

System.out.print((i + 1) + " ");

sum += (i + 1);

} else {

System.out.print("0 ");

}

} else {

System.out.println("Invalid Character");

return;

}

}

System.out.println();

}

sc.close();

System.out.println("Sum: " + sum);

}

}

**Output:**

Z

4

1 1 1 1

0 0 2 0

0 3 0 0

4 4 4 4

Sum: 25