import java.util.\*;

import java.util.concurrent.\*;

public class MatrixMultiplication {

// Method to multiply matrices using multithreading

public static int[][] multiplyMatrices(int[][] matrixA, int[][] matrixB) {

int rowsA = matrixA.length;

int colsA = matrixA[0].length;

int rowsB = matrixB.length;

int colsB = matrixB[0].length;

// Validate matrix dimensions

if (colsA != rowsB) {

throw new IllegalArgumentException("Matrix dimensions are not compatible for multiplication");

}

// Resultant matrix

int[][] result = new int[rowsA][colsB];

// Executor service to manage threads

ExecutorService executor = Executors.newFixedThreadPool(rowsA \* colsB);

List<Future<Void>> futures = new ArrayList<>();

// Submit tasks to calculate each element of the result matrix

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

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

final int row = i;

final int col = j;

futures.add(executor.submit(() -> {

result[row][col] = 0;

for (int k = 0; k < colsA; k++) {

result[row][col] += matrixA[row][k] \* matrixB[k][col];

}

return null;

}));

}

}

// Wait for all threads to complete

for (Future<Void> future : futures) {

try {

future.get();

} catch (InterruptedException | ExecutionException e) {

e.printStackTrace();

}

}

executor.shutdown();

return result;

}

// Helper function to print the matrix

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

for (int[] row : matrix) {

for (int elem : row) {

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

}

System.out.println();

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Get matrix A dimensions and elements from user

System.out.println("Enter number of rows and columns for matrix A:");

int rowsA = scanner.nextInt();

int colsA = scanner.nextInt();

int[][] matrixA = new int[rowsA][colsA];

System.out.println("Enter elements for matrix A:");

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

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

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

}

}

// Get matrix B dimensions and elements from user

System.out.println("Enter number of rows and columns for matrix B:");

int rowsB = scanner.nextInt();

int colsB = scanner.nextInt();

int[][] matrixB = new int[rowsB][colsB];

System.out.println("Enter elements for matrix B:");

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

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

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

}

}

// Multiply matrices using multithreading

try {

int[][] result = multiplyMatrices(matrixA, matrixB);

// Output the result

System.out.println("Result of the multiplication:");

printMatrix(result);

} catch (IllegalArgumentException e) {

System.out.println(e.getMessage());

}

}

}