Assignment 8 –

import java.util.Random;

public class ThreadedMatrix {

static final int MAX = 10;

static final int MAX\_THREAD = 10;

static int[][] matA = new int[MAX][MAX];

static int[][] matB = new int[MAX][MAX];

static int[][] matC = new int[MAX][MAX];

static int step\_i = 0;

static class Worker implements Runnable {

int i;

Worker(int i) {

this.i = i;

}

@Override

public void run() {

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

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

matC[i][j] += matA[i][k] \* matB[k][j];

}

}

}

}

public static void main(String[] args) {

Random rand = new Random();

// Generating random values in matA and matB

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

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

matA[i][j] = rand.nextInt(10);

matB[i][j] = rand.nextInt(10);

}

}

// Displaying matA

System.out.println("Matrix A");

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

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

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

}

System.out.println();

}

// Displaying matB

System.out.println("Matrix B");

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

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

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

}

System.out.println();

}

// declaring four threads

Thread[] threads = new Thread[MAX\_THREAD];

// Creating four threads, each evaluating its own part

long start = System.nanoTime();

for (int i = 0; i < MAX\_THREAD; i++) {

threads[i] = new Thread(new Worker(step\_i++));

threads[i].start();

}

// joining and waiting for all threads to complete

for (int i = 0; i < MAX\_THREAD; i++) {

try {

threads[i].join();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

long end = System.nanoTime();

System.out.println(end-start);

// Displaying the result matrix

// System.out.println("Multiplication of A and B");

// for (int i = 0; i < MAX; i++) {

// for (int j = 0; j < MAX; j++) {

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

// }

// System.out.println();

// }

}

}