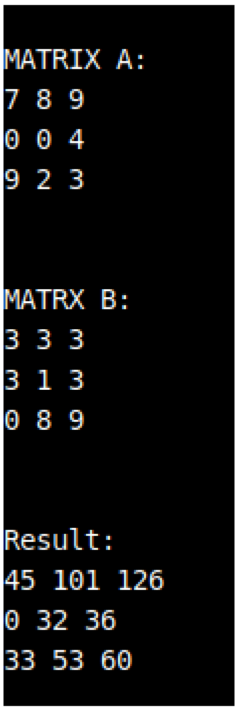
WAP TO IMPLEMENT MATRIX MULTIPLICATION USING THREADS

```
import java.util.Random;

public class Main{

//Creating the matrix
static int[][] mat = new int[3][3];
static int[][] mat2 = new int[3][3];
static int[][] result = new int[3][3];

public static void main(String [] args){

    //Creating the object of random class
    Random rand = new Random();


    //Filling first matrix with random values
    for (int i = 0; i < mat.length; i++) {
        for (int j = 0; j < mat[i].length; j++) {
            mat[i][j]=rand.nextInt(10);
        }
    }

    //Filling second matrix with random values
    for (int i = 0; i < mat2.length; i++) {
        for (int j = 0; j < mat2[i].length; j++) {
            mat2[i][j]=rand.nextInt(10);
        }
    }

    System.out.println("\n\nMATRIX A:");
    for (int i = 0; i < mat.length; i++) {
        for (int j = 0; j < mat[i].length; j++) {
            System.out.print(mat[i][j]+" ");
        }
        System.out.println();
    }

    System.out.println("\n\nMATRX B:");
    for (int i = 0; i < mat2.length; i++) {
        for (int j = 0; j < mat2[i].length; j++) {
            System.out.print(mat2[i][j]+" ");
        }
        System.out.println();
    }

    try{
        //Object of multiply Class
        Multiply multiply = new Multiply(3,3);
        //Threads
        MatrixMultiplier thread1 = new MatrixMultiplier(multiply);
        MatrixMultiplier thread2 = new MatrixMultiplier(multiply);
        MatrixMultiplier thread3 = new MatrixMultiplier(multiply);

        //Implementing threads
        Thread th1 = new Thread(thread1);
        Thread th2 = new Thread(thread2);
        Thread th3 = new Thread(thread3);

        //Starting threads
        th1.start();
        th2.start();
        th3.start();

        th1.join();
        th2.join();
        th3.join();

    }catch (Exception e) {
        e.printStackTrace();
    }

    //Printing the result
    System.out.println("\n\nResult:");
    for (int i = 0; i < result.length; i++) {
        for (int j = 0; j < result[i].length; j++) {
            System.out.print(result[i][j]+" ");
        }
        System.out.println();
    }
  }//End main

  }//End Class

   //Multiply Class
   class Multiply extends Main {

private int i;
private int j;
private int chance;

public Multiply(int i, int j){
    this.i=i;
    this.j=j;
    chance=0;
}

//Matrix Multiplication Function
public synchronized void multiplyMatrix(){

    int sum=0;
    int a=0;
    for(a=0;a<i;a++){
        sum=0;
        for(int b=0;b<j;b++){
            sum=sum+mat[chance][b]*mat2[b][a];
        }
        result[chance][a]=sum;
    }

    if(chance>=i)
        return;
    chance++;
}
}//End multiply class

//Thread Class
     class MatrixMultiplier implements Runnable {

private final Multiply mul;

public MatrixMultiplier(Multiply mul){
    this.mul=mul;
}


public void run() {
    mul.multiplyMatrix();
}
}
```



```
MATRIX A:
7 8 9
0 0 4
9 2 3


MATRX B:
3 3 3
3 1 3
0 8 9


Result:
45 101 126
0 32 36
33 53 60
```