Public Homework EEM 480 HW1 Homework 1

You are required to write a class, named `cMatrix` which generates `matrix` and the necessary functions are given below.. Don't use wrapper classes for primitive types. Use the primitive types in arrays. (Use `int` and `float` not `Integer` and `Float`)

```
public class cMatrix {
    private int [][] elements; // Keeps the elements of matrix in two
dimensional array
    private int row; // row size
    private int col; // column size

    public cMatrix(){ // Constructor generate matrix of size 10x10
    }

    public cMatrix(int row, int col){ //Constructor
    }

    public void AssignRandom(){ //Assigning random variables in range 1 -
10000
    }

    public void printMatrix(){ // Printing matrix in matrix form
    }

    public void printMatrixWithPrime(){ //Prints the matrix with "*" sign
near the prime elements
    }

    public cMatrix multiplyMatrices(cMatrix Multiplicand){ // Multiply two
matrices and informs the user about time lapse
    }
}
```

You can add your own methods to the cMatrix class.

When I import your class and try to run the following code :

```
public static void main(String[] args) {
    // TODO code application logic here
    cMatrix m1 = new cMatrix();
    m1.AssignRandom();
    m1.printMatrix();
    System.out.println("");
    m1.printMatrixWithPrime();
    cMatrix m2 = new cMatrix(10,20);
    m2.AssignRandom();
    cMatrix m3 = new cMatrix(20,10);
    m3.AssignRandom();
    cMatrix m4 = new cMatrix();
    m4 = m2.multiplyMatrices(m3);
    m4.printMatrixWithPrime();
}
```

The following output will be generated:

3146 8252 6214 5738 3535 5237 5835 7353 2751 5069

4715 3092 773 9055 3626 616 9528 9424 1510 4358

6832 5432 8581 3038 8354 4238 3452 1858 8102 9375

1105 6474 2698 5434 4104 7717 751 8107 4474 3766

5297 4306 7764 4146 2885 1599 1196 4900 5370 9325

2254 9243 8194 9624 6036 4546 5466 2444 5252 777

6981 235 4147 6469 1575 8043 5015 2049 7661 5786

9003 4446 3738 1742 7051 2639 8715 5949 7839 7066

5244 758 2885 5835 916 4251 1687 6235 5627 6812

7419 9400 2285 7792 4025 2322 4134 4066 9783 2242

3146 8252 6214 5738 3535 5237* 5835 7353 2751 5069

4715 3092 773* 9055 3626 616 9528 9424 1510 4358

6832 5432 8581* 3038 8354 4238 3452 1858 8102 9375

1105 6474 2698 5434 4104 7717* 751* 8107 4474 3766

5297* 4306 7764 4146 2885 1599 1196 4900 5370 9325

2254 9243 8194 9624 6036 4546 5466 2444 5252 777

6981 235 4147 6469* 1575 8043 5015 2049 7661 5786

9003 4446 3738 1742 7051 2639 8715 5949 7839 7066

5244 758 2885 5835 916 4251 1687 6235 5627 6812

7419 9400 2285 7792 4025 2322 4134 4066 9783 2242

The duration of multiplication of matrices 10x20 * 20x10 is 0 millisecond

531276823 486624792 562001563* 450702641 506533036 460796033 471205373 530611889* 439201220 500631314

498381550 434758568 440492928 512120190 493367257* 403792849 476866878 464320813 382008660 426918882

448194951 409147481* 481006668 428325198 427569264 434553297 511477231 435163389 288536713 420779528

608149120 469341351 531353084 563094472 533332146 441727036 551226830 474137687* 419779495 452437080

442794355 511319794 530249965 440296790 476315641 436206646 510150108 485292025 453195005 498092641

585342902 550948593 556822515 591489241 564034739 567320327 609549820 620859193 518708268 557903891

541035337* 510074070 611761340 580707539 545134105 525630590 526247090 515458181 448292732 516665752

326958842 354718128 374449628 341690196 347609971 383719995 337541586 358870302 343601639 379875878

444587044 403945083 491979257 457586030 431006630 399118242 359173207 398266984 377680144 403179225

490556163 417620363* 484118292 494137785 471346131 390006236 457053316 394350949* 395402313 429408476

BUILD SUCCESSFUL (total time: 18 seconds)

When I import your class and run the following code :

```
public static void main(String[] args) {
    // TODO code application logic here
    cMatrix m2 = new cMatrix(1000,1000);
    m2.AssignRandom();
    cMatrix m3 = new cMatrix(1000,1000);
    m3.AssignRandom();
    cMatrix m4 = new cMatrix();
    m4 = m2.multiplyMatrices(m3);
}
```

The following output will be generated:

```
The duration of multiplication of matrices 1000x1000 * 1000x1000 is 6592 millisecond
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
Here another output :
The duration of multiplication of matrices 1500x1500 * 1500x1500 is 30358 millisecond
BUILD SUCCESSFUL (total time: 30 seconds)

Here another output :
The duration of multiplication of matrices 2000x2000 * 2000x2000 is 84590 millisecond
BUILD SUCCESSFUL (total time: 1 minute 24 seconds)
```

In your report, draw the graph of time vs matrix size in different size square matrix multipkications.

```
Test your multiplication with :
10x10
20x20
50x50
100x100
200x200
500*500
750x750
1000x1000
1200x1200
1500x1500
1750x1750
2000x2000
```

- Search and find "`random`" methods to fill the arrays with random numbers.
- Search and find how to measure the running time in milliseconds.
- The utmost matrix limit is 10000 x 10000
- Do them by yourself.

Rules for HW Submission

. You have to write your HW in NetBeans environment.
. You have to write a report with name "**Report_HW1.pdf**" explaining your HW (purpose, how did you solve it, algorithm etc.) and what you the environment you used (NetBeans, for example). The person who read your report can easily use the class you have written.
. Discuss the result you have obtained.
. Submission should be in the form of a zip. When extracted, the result should be a single folder with the name "HW1".
. Don't forget to put your report into the zip file.
. The name of your project will be "**Name_Surame_HW1**. e.g. *Lutfullah_Arici_HW1*. If you do not obey the rule I will not grade your homework.
. You have to bundle your whole project folder into your HW1.zip file.
. If I extract your project file, then import to my environment and if it doesn't work, you will be graded on 30 not 100. (Double check. It saves life)
. Do HW by yourself. Be honest.