

Name: Le Trung Hieu
Student's ID: 1852365

Exercise 1: Posix thread

Matrix 10x10

```
[student@ccc02ip95 lab1]$ time ./a.out 10
Yup, we're good

real    0m0.005s
user    0m0.000s
sys     0m0.009s
[student@ccc02ip95 lab1]$
```

Matrix 100x100

```
[student@ccc02ip95 lab1]$ time ./a.out 100
Yup, we're good

real    0m0.004s
user    0m0.011s
sys     0m0.004s
[student@ccc02ip95 lab1]$ █
```

Matrix 1000x1000

```
[student@ccc02ip95 lab1]$ time ./a.out 1000
Yup, we're good

real    0m0.485s
user    0m10.557s
sys     0m0.011s
[student@ccc02ip95 lab1]$ █
```

Matrix 10000x10000

```
[student@ccc02ip95 lab1]$ time ./a.out 10000
Yup, we're good

real    23m55.433s
user    440m3.118s
sys     0m1.542s
[student@ccc02ip95 lab1]$ █
```

OpenMP

Matrix 10x10

```
[student@ccc02ip95 lab1]$ time ./matrix 100
Carrying out matrix-matrix multiplication
Checking the results...
Yup, we're good!
Computing time: 0.060831

real    0m0.352s
user    0m0.202s
sys     0m0.003s
[student@ccc02ip95 lab1]$
```

Matrix 100x100

```
[student@ccc02ip95 lab1]$ time ./matrix 1000
Carrying out matrix-matrix multiplication
Checking the results...
Yup, we're good!
Computing time: 2.021079

real    0m2.182s
user    0m18.164s
sys     0m0.014s
[student@ccc02ip95 lab1]$
```

Matrix 1000x1000

```
[student@ccc02ip95 lab1]$ time ./matrix 10
Carrying out matrix-matrix multiplication
Checking the results...
Yup, we're good!
Computing time: 0.067494

real    0m0.082s
user    0m0.184s
sys     0m0.003s
[student@ccc02ip95 lab1]$
```

Matrix 10000x10000

```
[student@ccc02ip95 lab1]$ time ./matrix 10000
Carrying out matrix-matrix multiplication
Checking the results...
Yup, we're good!
Computing time: 1413.865642

real    23m39.310s
user    477m9.413s
sys     0m1.786s
[student@ccc02ip95 lab1]$
```

Table:

	Posix thread	OpenMP
Matrix 10x10	0m0.005s	0m0.352s
Matrix 100x100	0m0.004s	0m2.182s
Matrix 1000x1000	0m0.485s	0m0.082s
Matrix 10000x10000	23m55.433s	23m39.310s

Conclude: The Posix thread is faster than the OpenMP for the small size of matrix. When the size of matrix is bigger, the OpenMP method seems to behave better.

Exercise 2:

```
[student@cc02ip95 lab1]$ time OMP_NUM_THREADS=48 ./pi 100000000000000  
Count = 217125024, Samples = 276447232, Estimate of pi: 3.1416487324  
Actual pi: 3.14159265359  
real    0m0.664s  
user    0m10.609s  
sys     0m0.013s  
[student@cc02ip95 lab1]$
```

For 100000000000000 samples, the program still can run very fast but the result seems to be not improved when increase the number of samples. It may because the random data I have created is not good.