2. In this problem you will develop two different implementations of the computation of **`pi`** numerically using pthreads and OpenMP. Then you will compare them in terms of scalability. In this problem, you will develop a program that computes the value of **`pi`**.

System: Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
CPUs: 28
Sockets: 2
Cores per socket: 14
Threads per core: 1

a. Evaluate the speedup that you achieve by using pthreads and multiple cores. You are free to use as many threads as you like. The program should take two input parameters, the number of threads and the number of "darts" thrown. Your program should print out the time required to compute pi and the final value of pi. Make sure to document the system you are running on and the number of hardware threads available.

Number of darts: 1, 000, 000, 000
Number of threads: 1, 2, 4, 8, 16, 28

```
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
1000000000
number of threads:
1
total time is: 14.740000s
3.14159254399999987584
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
1000000000
number of threads:
2
total time is: 14.730000s
3.14160771199999988568
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
1000000000
number of threads:
4
total time is: 14.710000s
3.14158996799999989946
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
1000000000
number of threads:
8
total time is: 14.710000s
3.14160777599999985199
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
1000000000
number of threads:
16
total time is: 14.580000s
3.14168588800000003758
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
1000000000
number of threads:
28
total time is: 14.700000s
3.14179588799999987003
```

b. Now develop the same program using OpenMP. Repeat all of the steps requested in part a).

Number of darts: 1, 000, 000, 000
Number of threads: 1, 2, 4, 8, 16

```
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
1000000000
number of threads:
1
3.14164729599999992260
total time is: 19.150000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
1000000000
number of threads:
2
3.14166107600000010791
total time is: 19.130000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
1000000000
number of threads:
4
3.14164221600000015400
total time is: 19.130000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
1000000000
number of threads:
8
3.14151178399999997382
total time is: 19.370000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
1000000000
number of threads:
16
3.14157846400000018150
total time is: 19.140000s
```

c. Now compare the two implementations in terms of strong and weak scaling, where the number of Monte Carlo simulations (i.e., "darts" thrown) is used to assess weak scaling.

According to the formula: $Speedup = t(1)/t(N)$
For Strong Scaling, If the amount of time needed to complete a serial task t(1), and the amount of time to complete the same unit of work with N processing elements (parallel task) is t(N). For Weak Scaling, If the amount of time to complete a work unit with 1 processing element is t(1), and the amount of time to complete N of the same work units with N processing elements is t(N). (Cited from https://hpc-wiki.info/hpc/Scaling_tutorial)

```
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
2000000000
number of threads:
2
total time is: 29.350000s
3.14159254399999987584
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
4000000000
number of threads:
4
total time is: 58.320000s
3.14159254399999987584
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
8000000000
number of threads:
8
total time is: 116.610000s
3.14159254399999987584
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
16000000000
number of threads:
16
total time is: 233.550000s
3.14159254399999987584
[luo.qiu@c2177 hwtest]$ ./p
number of darts:
28000000000
number of threads:
28
total time is: 407.670000s
3.14159254399999987584
```

Pthreads:

Threads	1	2	4	8	16	28
Strong Scaling	100%	100.07%	100.20%	100.20%	101.10%	100.27%
Weak Scaling	100%	50.22%	25.27%	12.64%	6.31%	3.62%

```
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
2000000000
number of threads:
2
3.14159602199999987704
total time is: 38.190000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
4000000000
number of threads:
4
3.14160980800000011470
total time is: 76.330000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
8000000000
number of threads:
8
3.14162639400000021084
total time is: 152.640000s
[luo.qiu@c2177 hwtest]$ ./o
number of darts:
16000000000
number of threads:
16
3.14158244874999992291
total time is: 305.810000s
```

OpenMP:

Threads	1	2	4	8	16
Strong Scaling	100%	100.10%	100.10%	98.86%	100.05%
Weak Scaling	100%	50.14%	25.09%	12.55%	6.26%