

Exercise 12. Answer Sheet

Student's Name: ____Tran Thi Thoa____

Student's ID: ____s1242006____

Problem 1. (40 points) Implement the randomized algorithm for calculating π given in the lecture.

- Calculate π using your program 10 times using N number of points from the next table. Fill in the results you got.

N	1	2	3	4	5	6	7	8	9	10
1	4	4	4	4	4	4	4	4	4	4
10	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2
100	2.92	2.92	2.92	2.92	2.92	2.92	2.92	2.92	2.92	2.92
1000	3.164	3.164	3.164	3.164	3.164	3.164	3.164	3.164	3.164	3.164
10000	3.1472	3.1472	3.1472	3.1472	3.1472	3.1472	3.1472	3.1472	3.1472	3.1472
100000	3.14408	3.14408	3.14408	3.14408	3.14408	3.14408	3.14408	3.14408	3.14408	3.14408

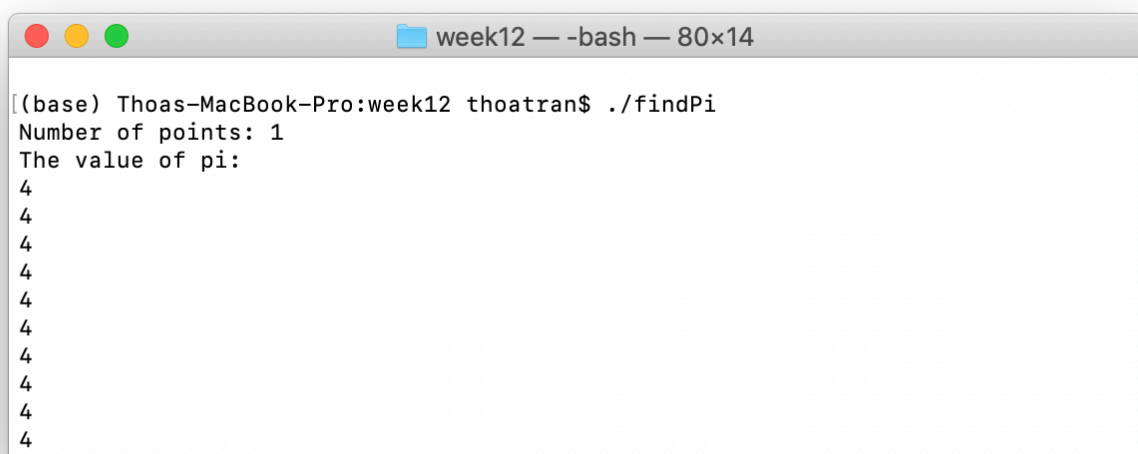
- Upload your code.

The implementation is on the file findPi.cpp

To compile and run the program, run the following command lines:

```
g++ -std=c++11 -o findPi findPi.cpp  
./findPi
```

The output for N = 1, 10, 100, 1000, 10000, 100000 respectively:



```
week12 — -bash — 80x14  
[(base) Thoas-MacBook-Pro:week12 thoatran$ ./findPi  
Number of points: 1  
The value of pi:  
4  
4  
4  
4  
4  
4  
4  
4  
4  
4  
4
```

```
week12 — -bash — 80×14
[(base) Thoas-MacBook-Pro:week12 thoatran$ ./findPi
Number of points: 10
The value of pi:
3.2
3.2
3.2
3.2
3.2
3.2
3.2
3.2
3.2
3.2
```

```
week12 — -bash — 80×14
[(base) Thoas-MacBook-Pro:week12 thoatran$ ./findPi
Number of points: 100
The value of pi:
2.92
2.92
2.92
2.92
2.92
2.92
2.92
2.92
2.92
2.92
```

```
week12 — -bash — 80×14
[(base) Thoas-MacBook-Pro:week12 thoatran$ ./findPi
Number of points: 1000
The value of pi:
3.164
3.164
3.164
3.164
3.164
3.164
3.164
3.164
3.164
3.164
```

```
week12 — -bash — 80×14

[(base) Thoas-MacBook-Pro:week12 thoatran$ ./findPi
Number of points: 10000
The value of pi:
3.1472
3.1472
3.1472
3.1472
3.1472
3.1472
3.1472
3.1472
3.1472
3.1472
```

```
week12 — -bash — 80×14

[(base) Thoas-MacBook-Pro:week12 thoatran$ ./findPi
Number of points: 100000
The value of pi:
3.14408
3.14408
3.14408
3.14408
3.14408
3.14408
3.14408
3.14408
3.14408
3.14408
```

Problem 2. (60 points) Write a program implementing the quicksort algorithm. Make two versions:
a) Randomized quicksort, where the pivot element is chosen at random.
b) Deterministic quicksort, where the pivot element is always the first element of the array.

- Generate random sequence of length N (as given in the Table)
- Measure the time each quicksort version needs to sort the sequences (fill in the average of 100 runs).

N	100	1000	10000	100000	1000000
Randomized	5.14	105.28	1301.16	21846	1.08873E+06
Deterministic	4.26	105.56	1325.89	22132.3	1.08776E+06

All the durations above are in microsecond

- Upload your code.

The implementations of quick sort with the first element as pivot and choosing a random element as pivot are in the file quickSortRand.cpp

To compile and run the file, run the following command line:

```
g++ -std=c++11 -o quickSortRand quickSortRand.cpp  
./quickSortRand
```

The output will be like:

```
week12 — -bash — 80x24  
quickSortRand.cpp:125:58: error: use of undeclared identifier 'start2'  
    duration2 += duration_cast<microseconds>(stop2 - start2).count();  
                                                    ^  
8 errors generated.  
[(base) Thoas-MacBook-Pro:week12 thoatran$ g++ -std=c++11 -o quickSortRand quickSortRand.cpp]  
[(base) Thoas-MacBook-Pro:week12 thoatran$ ./quickSortRand ]  
-----N = 100-----  
The time for quick sort with the first element as pivot: 4.26microseconds  
The time for quick sort with the random element as pivot: 5.14microseconds  
-----N = 1000-----  
The time for quick sort with the first element as pivot: 105.56microseconds  
The time for quick sort with the random element as pivot: 105.28microseconds  
-----N = 10000-----  
The time for quick sort with the first element as pivot: 1325.89microseconds  
The time for quick sort with the random element as pivot: 1301.16microseconds  
-----N = 100000-----  
The time for quick sort with the first element as pivot: 22132.3microseconds  
The time for quick sort with the random element as pivot: 21846microseconds  
-----N = 1000000-----  
The time for quick sort with the first element as pivot: 1.08876e+06microseconds  
The time for quick sort with the random element as pivot: 1.08773e+06microseconds  
s  
(base) Thoas-MacBook-Pro:week12 thoatran$ █
```