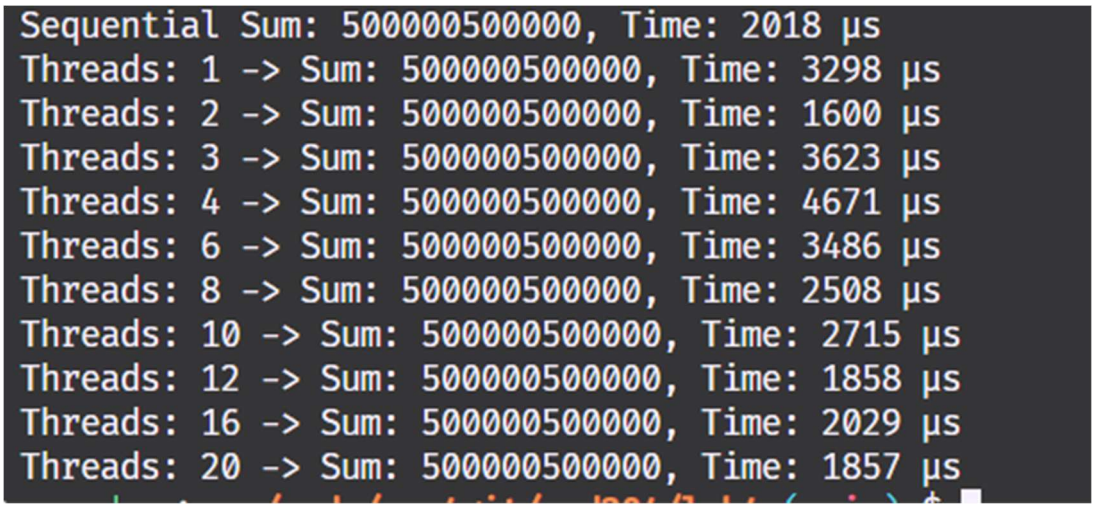
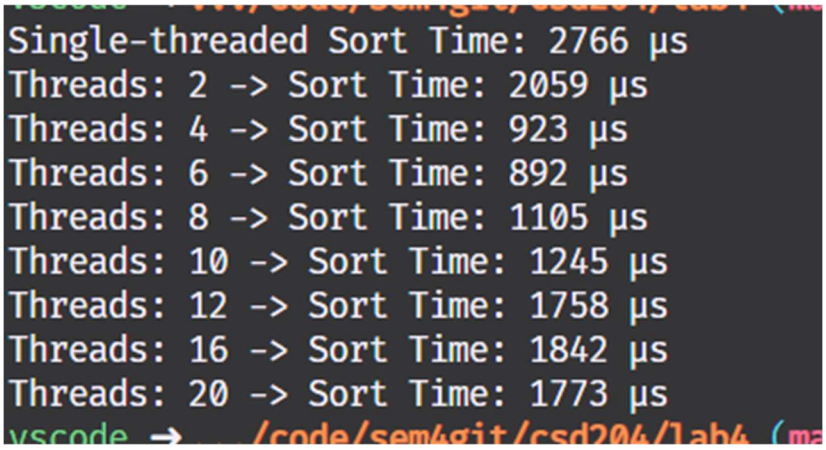
Shreyas Marwah
2310110604

Question 1



The sequential sum takes 2018 µs, while using one thread takes 3298 µs, making it slower due to thread creation overhead. With two threads, the time drops to 1600 µs, showing the best speedup. Three and four threads take 3623 µs and 4671 µs, likely due to uneven workload distribution. Performance varies as more threads are added. Twelve and twenty threads take 1858 µs and 1857 µs, but adding more threads does not always help. Too many threads increase overhead, reducing efficiency. The best result comes with two threads, while higher counts show diminishing returns due to CPU limitations.

Q2



The single-threaded sort took 2766 µs, while multithreading improved speed initially. With 2 threads, the time dropped to 2059 µs, and 4 threads gave the best performance at 923 µs. Using 6 threads, the time was 892 µs, slightly better. However, with more threads, performance fluctuated. 8 and 10 threads took 1105 µs and 1245 µs, showing diminishing returns. Beyond 12 threads, performance worsened, reaching 1842 µs at 16 threads and 1773 µs at 20 threads. This suggests overhead from thread management outweighs the speed gains beyond a certain point, making 4 to 6 threads optimal.

Q3

Shreyas Marwah
2310110604

```
● vscode ➜ .../code/sem4git/csd204/lab4 (main) $ g++ code3.cpp -pthread
● vscode ➜ .../code/sem4git/csd204/lab4 (main) $ ./a.out
  Without Lock -> Counter: 1258750, Time: 16851 µs
  With Lock -> Counter: 2000000, Time: 46346 µs
```

The without lock version resulted in a counter value of 1,258,750 instead of the expected 2,000,000, showing that many increments were lost due to race conditions. However, it was faster (16,851 µs) since there was no synchronization overhead. In contrast, the with lock version correctly reached 2,000,000, ensuring accuracy, but it was slower (46,346 µs) due to the mutex operations. This shows that locks prevent data corruption but add execution overhead, making performance a tradeoff between speed and correctness.

the graphs given below are obtained when plotting the data in q1 (blue) and q2 (red)