**PC - Lab 04 IT23226128**

**Exercise 01**

1. **Compile and run your program using the command below**

gcc -fopenmp lab04a.c -o lab04a.o

./lab04a.o

1. **Is the answer correct now?**

Yes, the answers are now correct. The calculated value of π is approximately 3.141592653589, which is accurate to 12 decimal places.

1. **Compare the speeds of the original program you wrote in Lab03 with data races happening and when you use a mutex (critical section, atomic section)**

The timing data clearly shows the performance impact of different synchronization methods:

* output\_nosync.txt (Data Race - Incorrect): 2.224 seconds
* output1\_atomic.txt (Atomic - Correct): 1.965 seconds
* output1\_critical.txt (Critical - Correct): 0.147 seconds
* output\_serial.txt (Serial - Correct): 1.678 seconds

The critical section was the fastest, which is unusual as atomic is typically faster for simple operations. This suggests the hardware-level atomic operation had very high overhead for this specific case.

This overhead was so severe that the atomic version was actually slower than the single-threaded serial program, completely negating the benefit of parallelization.

The unsynchronized version with a data race was the slowest due to "cache thrashing," where cores constantly fight over the shared variable. This shows data races are both incorrect and terrible for performance.

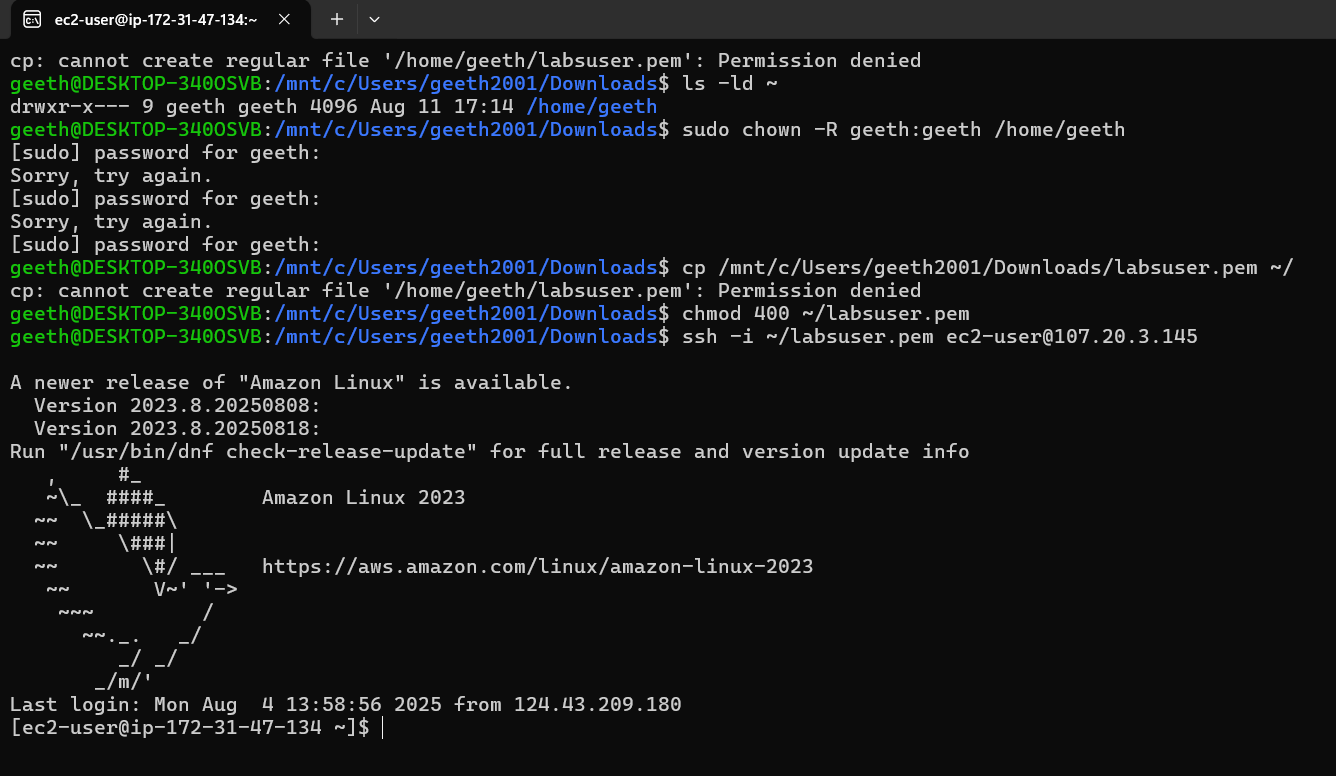
1. **How many cores does your linux machine have?**

* 2 cores

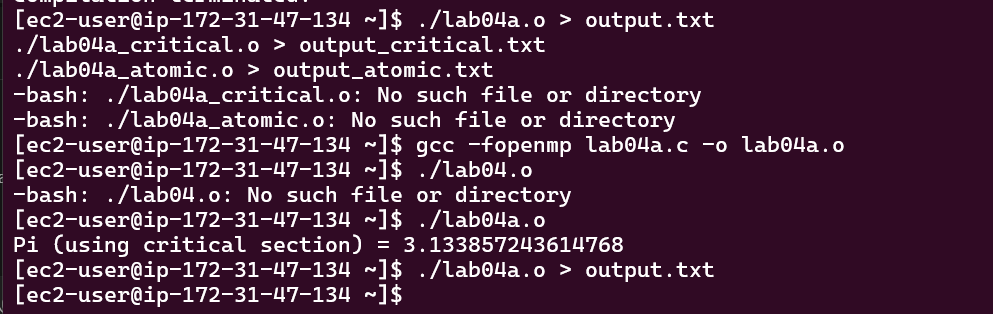
1. **Use an new EC2 configuration that is higher tryout the program and change the number of threads used (See Lab02), is there a difference when you run the serial program and the parallel program in this higher EC2 configuration?**

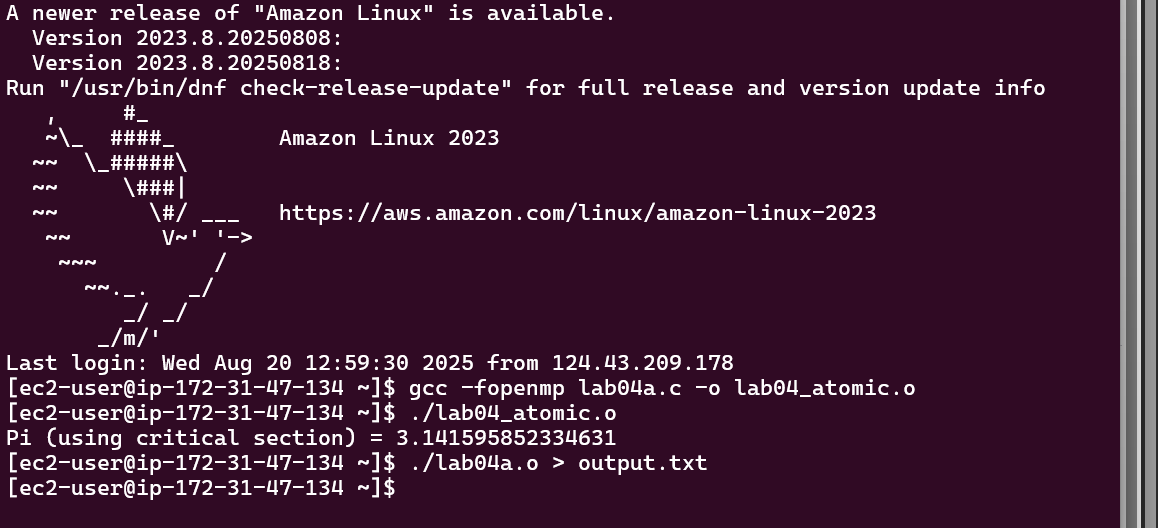
Yes, there will be a significant difference. The serial program will run faster due to a better CPU. The parallel program (especially with critical) will show much greater speedup when using more threads, as there are more physical cores to execute them concurrently, reducing the parallel time drastically.

1. **What is the speedup for both EC2 instances when you change number of threads in this new version of the program**

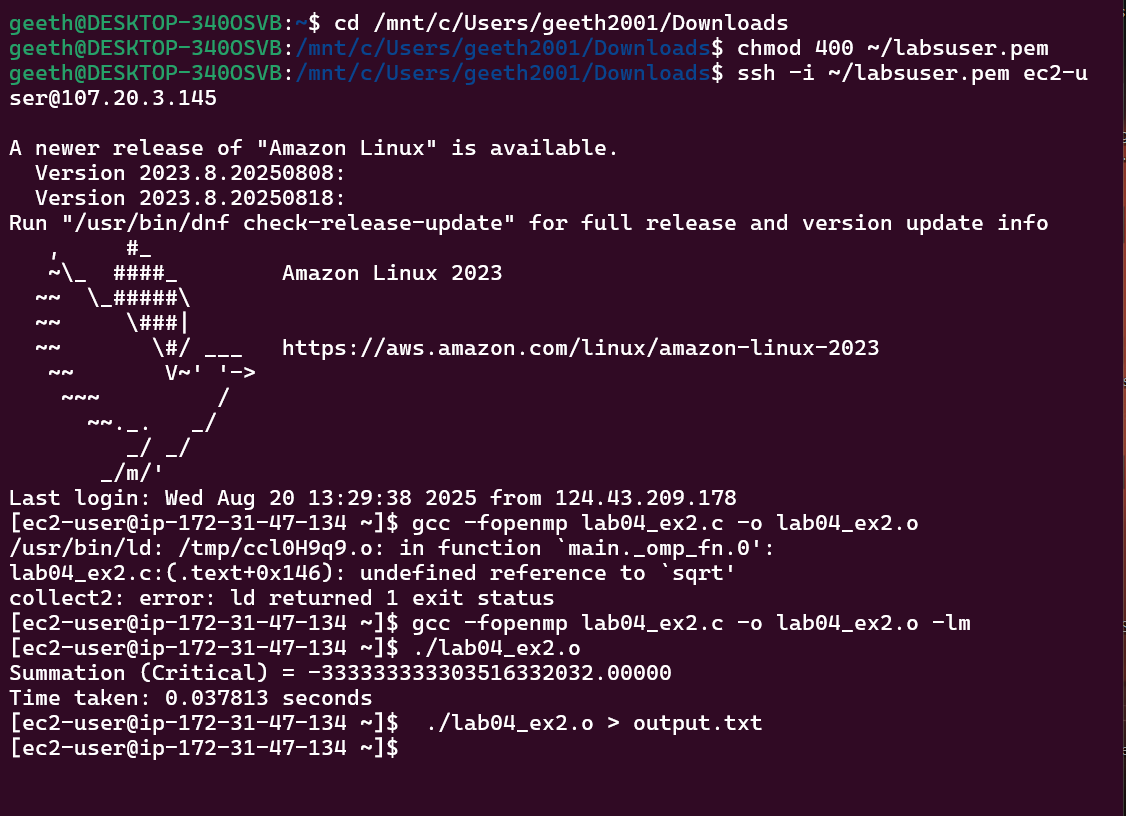
****

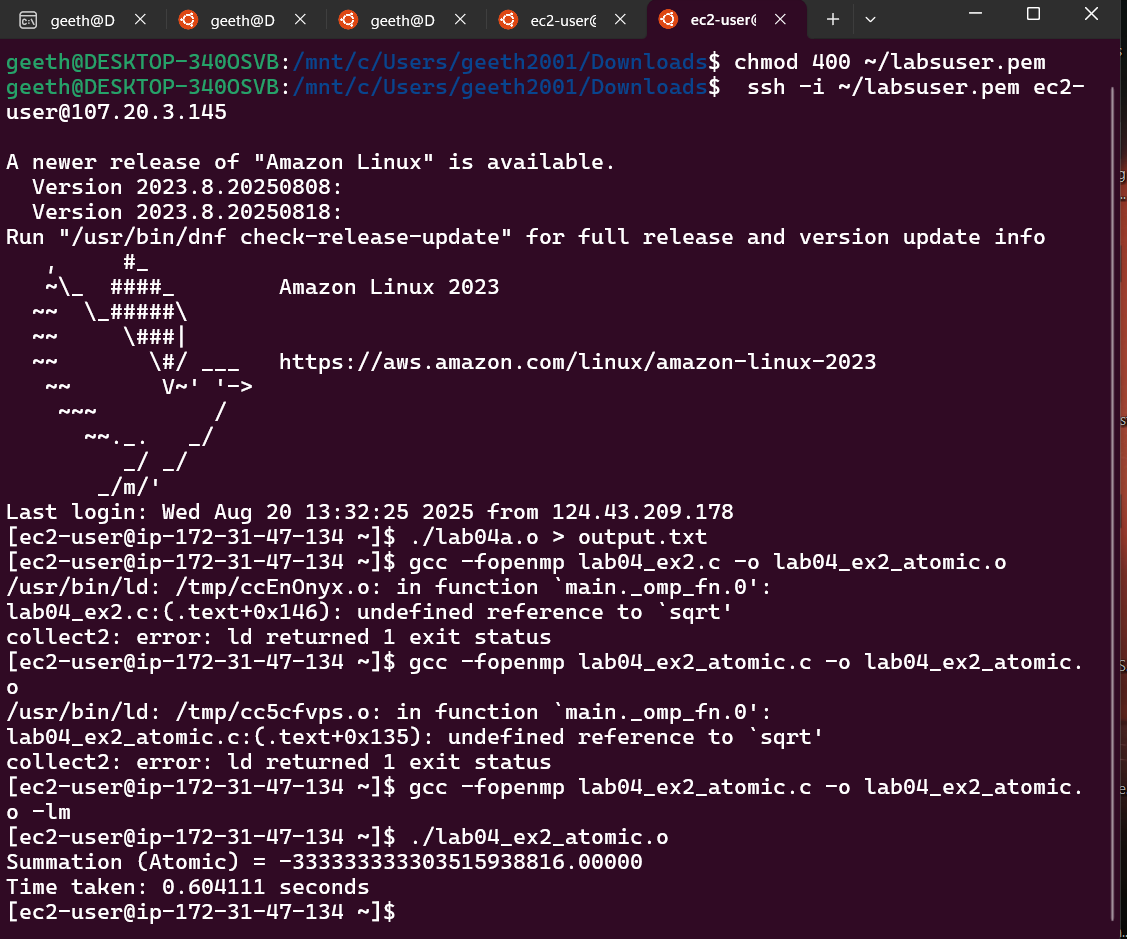
****

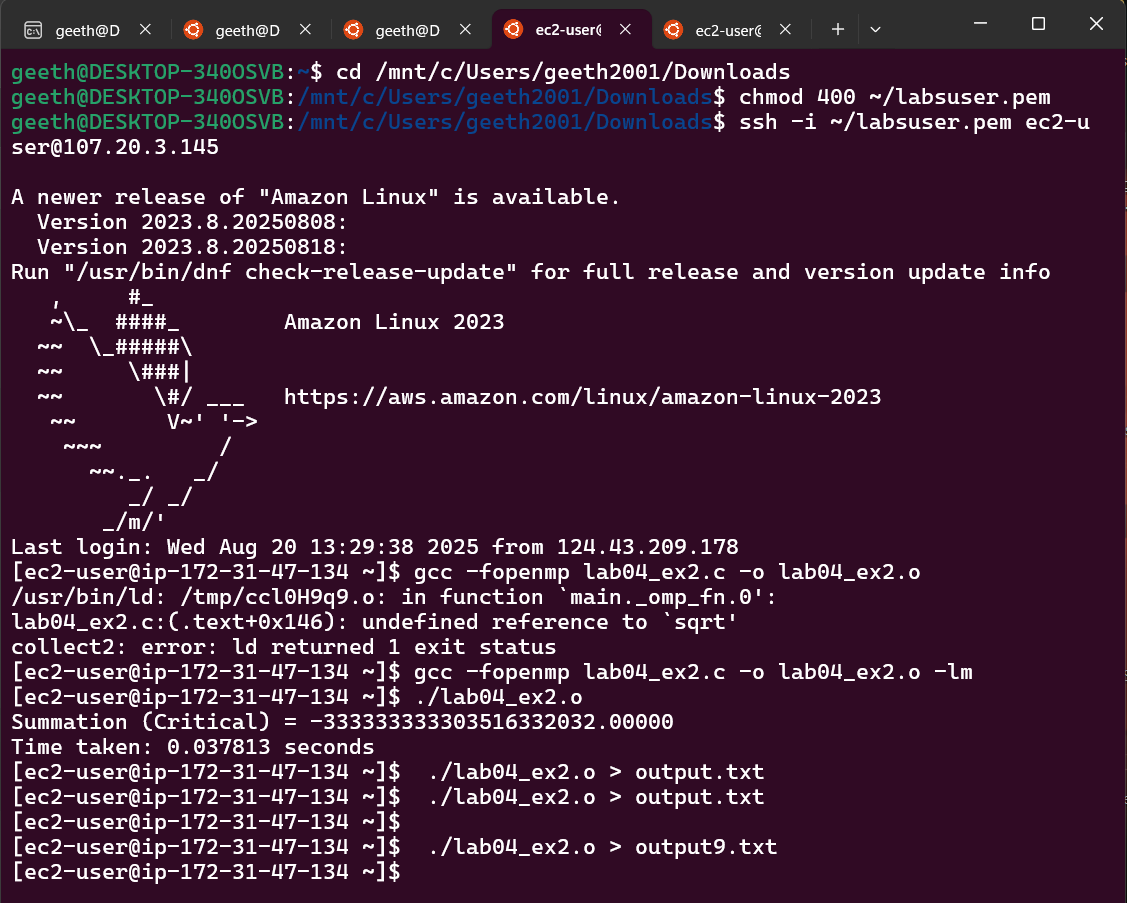


****

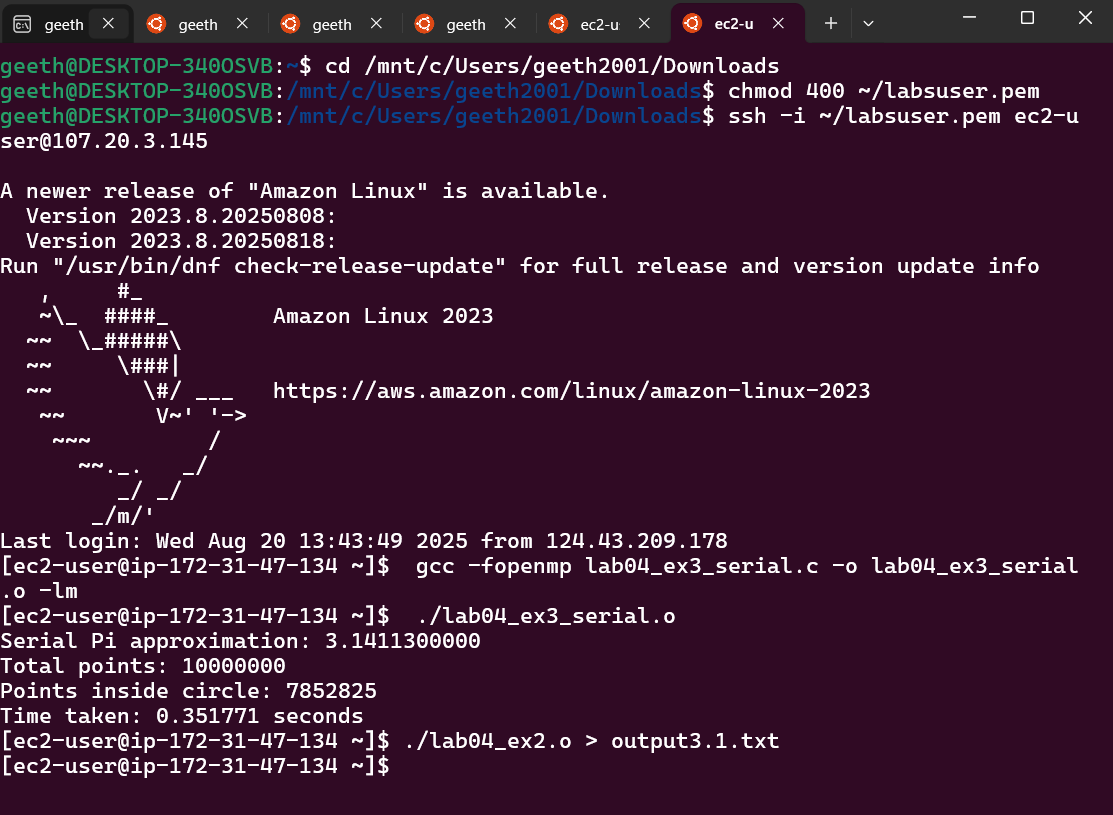
**Exercise 02**

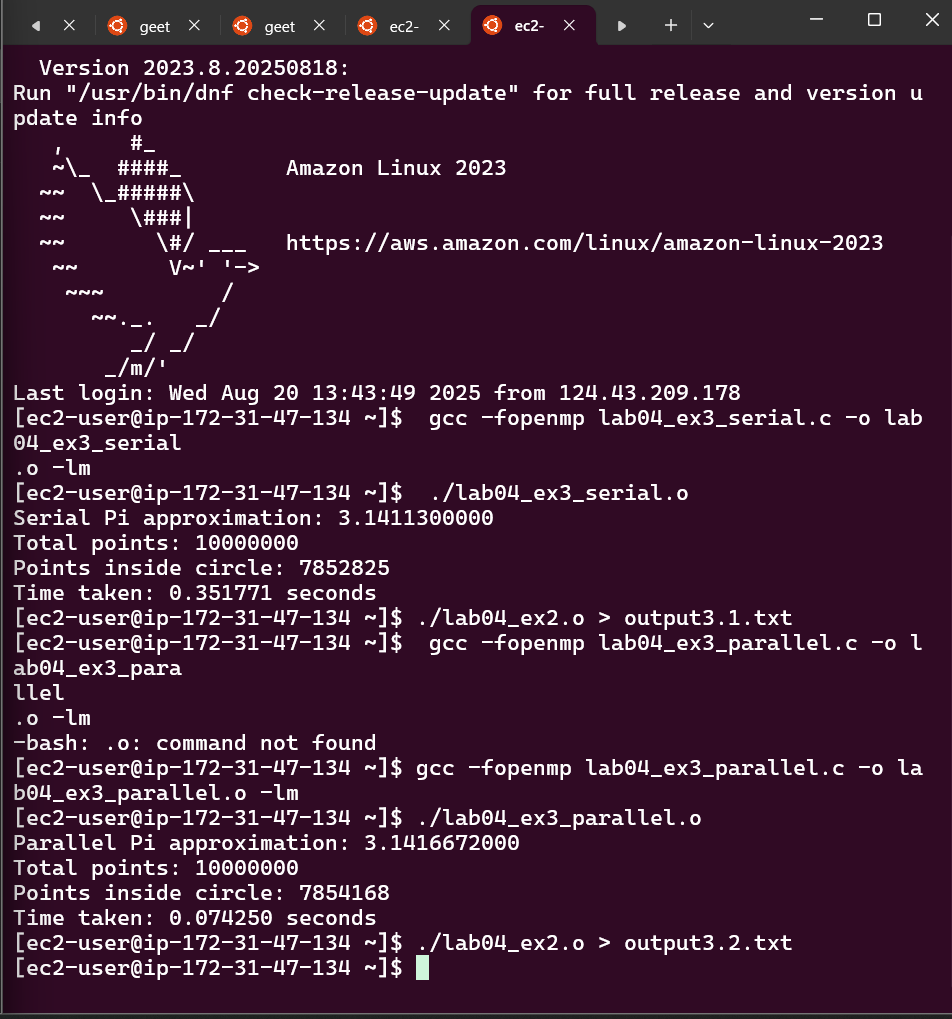
****

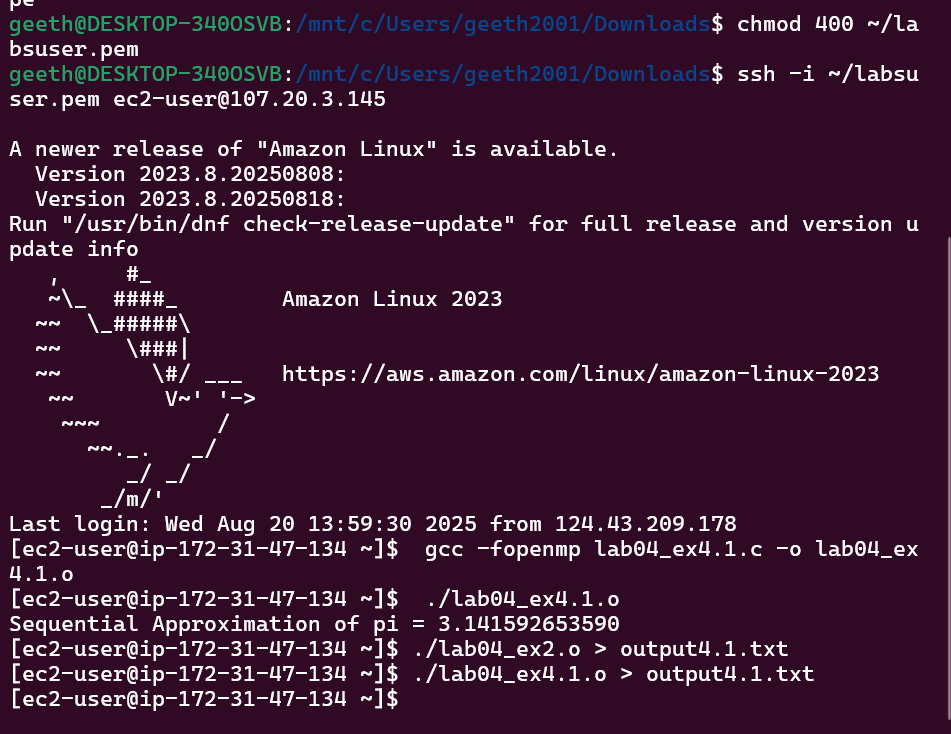


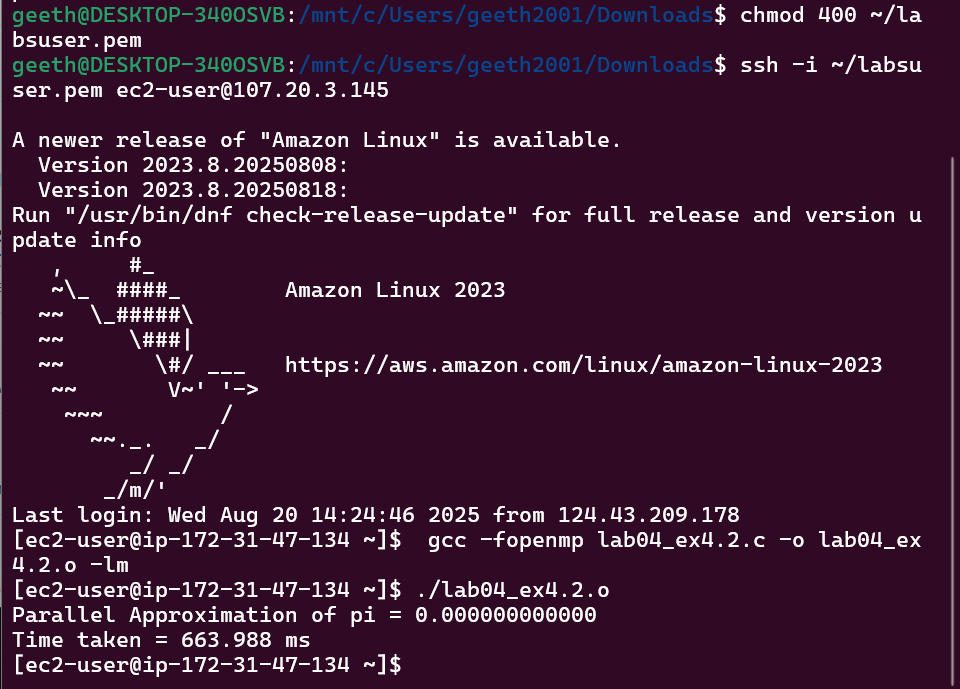


**Exercise 03**

****

****

**Exercise 04**

****