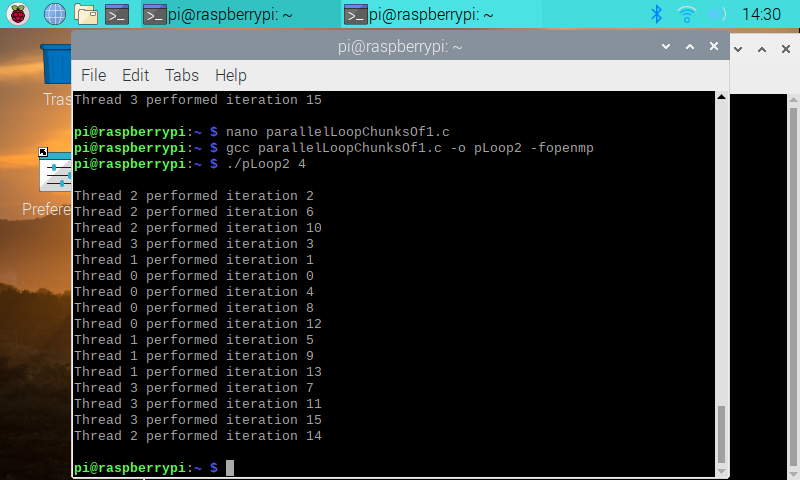
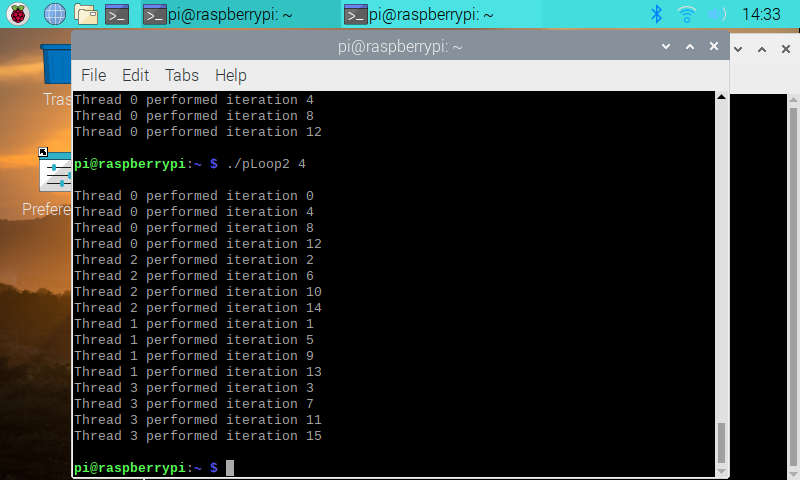
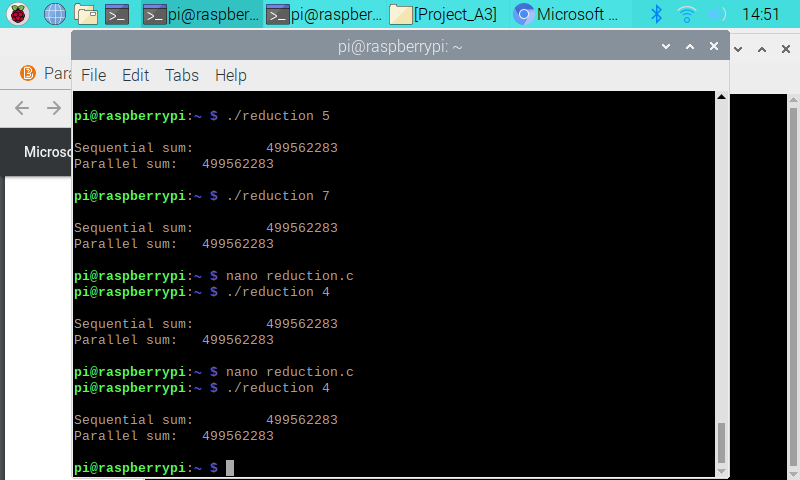
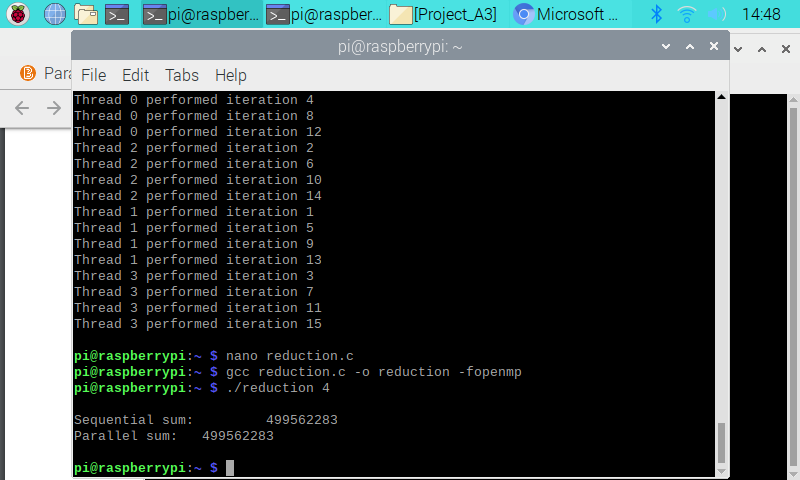
Bryanna Hardy

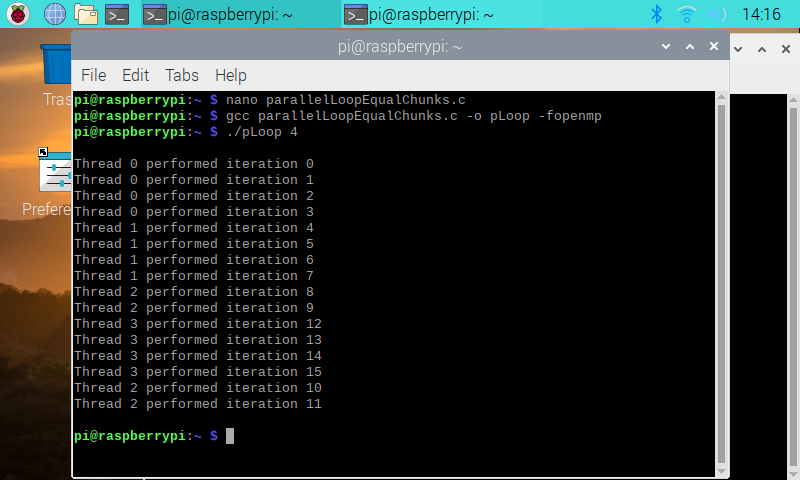
Task 3: Parallel Programming Basics

During this tutorial program, I have noticed that the iterations come out of order every time you run the code, instead of sequentially. In addition, you can see that the iterations correspond to the correct thread number. Also, because there are normally 4 threads in the Raspberry PI, each thread with iterate 4 times each, as shown above in the image. It is referred to as the data decomposition pattern because you are decomposing the amount of work that needs to be done across multiple threads.



In this program, as it went through each loop the sequential and parallel sum never changed. I even changed the number of threads to see if it would change but it never did. I guess the reason why is because each accumulator needs to be private to each thread ofits own to produce the correct answer. In this type, the variable sum is dependent on what all the other threads are doing to compute.



.

In this tutorial program, each thread is in order based on the iteration numbers. They were computed in equal chunks. Basically, the work was split evenly to compute. Once one iteration of the loop is completed, it does to the next thread, and so on. Also, it is done statically, so9 that each thread has an equal amount of work, not consecutive iterations.