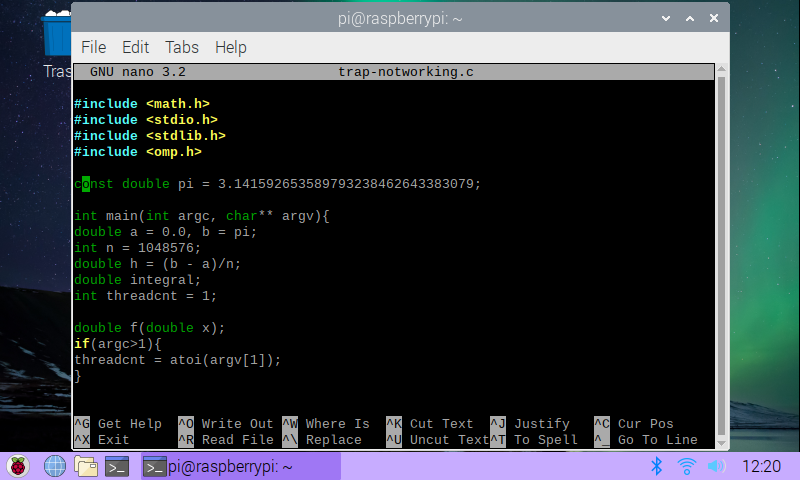
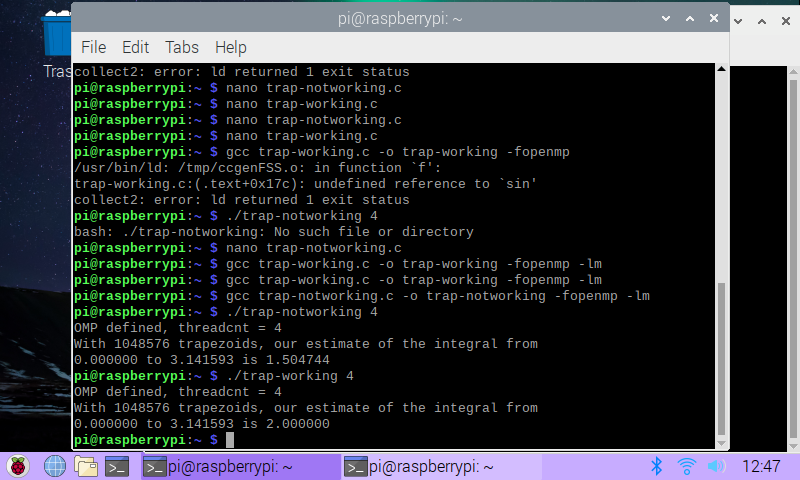
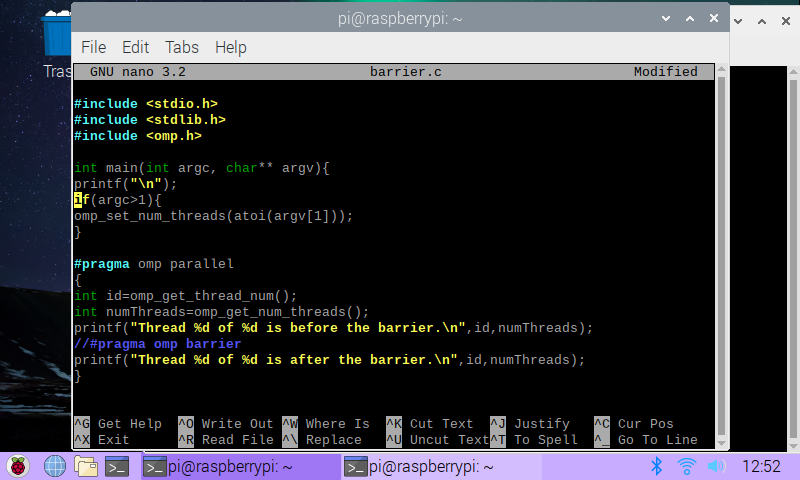
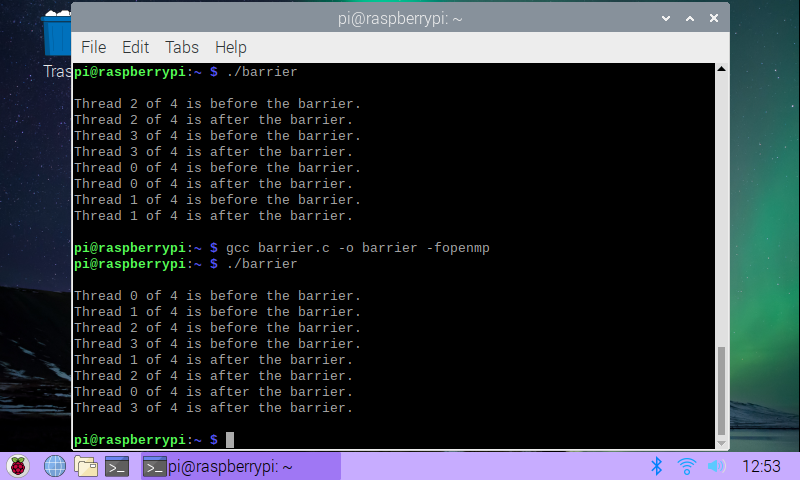
Zoe Kosmicki

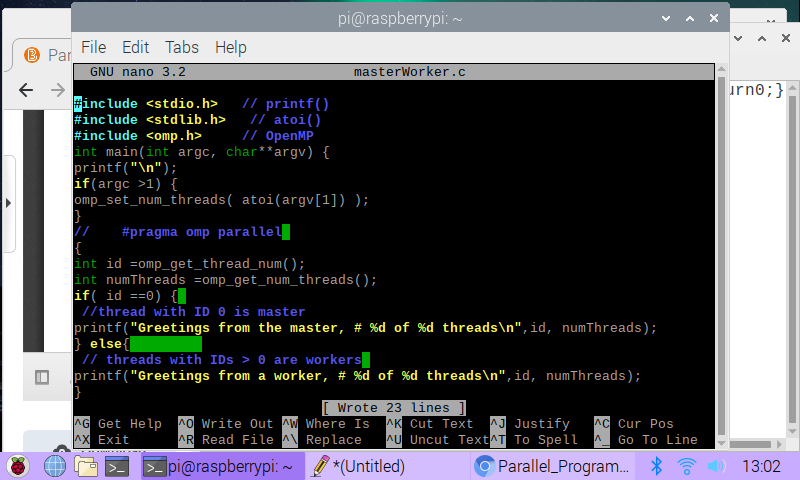
Assignment 4

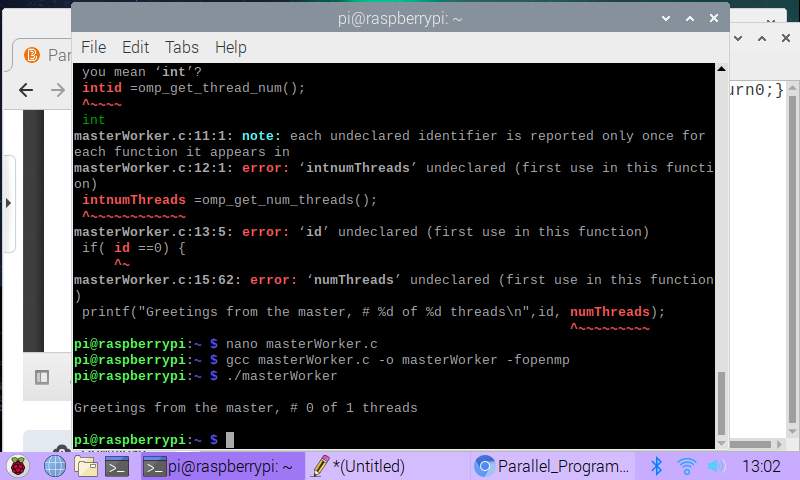
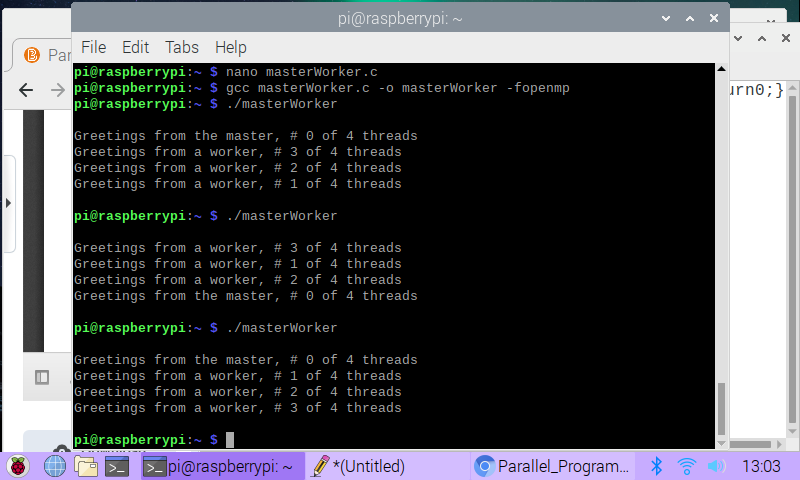
Task 3 Part B

First I wrote out the code for trap-notworking.c and trap-working.c.

After fixing a few spelling errors and an error having to do with linking the math library, I created executables from both files and ran both with the input 4.As expected, the notworking file gives an incorrect answer, while the working file gives the correct one.

Next I wrote out the code for barrier.c, starting with the barrier pragma commented out.Afterwards, I created the executable file and ran it as-is, and then with the barrier pragma uncommented, following the same steps. Without the barrier, the threads have no set order to execute in, so we see the befores and afters being mixed up. With the barrier uncommented, we can see that every thread before and after the barrier are all grouped together as intended. 

Next, I wrote out the code for masterWorker.c, shown below.With the parallel pragma commented out, I linked and ran the executable, and got the output shown below.

This makes sense, since master-worker is a type of parallel pattern, so if the program isn’t running in parallel, there’s only the master thread and no workers.After uncommenting the parallel pragma and running the program again a few different times, we can see the master thread appears once, and each worker thread having their unique threads executed successfully.