Lord of the Rings
Programming Project 1 (minor – 30 points)
due 11 PM Friday, 27 January 2023

A few days have passed in the Shire since Frodo (stupid Frodo) left with the foolish wizard Gandalf to cast (i.e., drop) the One Ring in the fiery lava pits of Mount Doom. Frankly, the whole "quest" sounded kind of dumb. I mean, couldn't someone melt down the ring anywhere? Plus, word on the street is that Gandalf could just summon giant eagles to fly straight to Mordor. God knows why they're walking. They didn't even look for decent fares on Amtrak. Losers. All of them.

You personally don't carry any bitterness over the fact that you were overlooked for the big dumb quest. It also doesn't bother you that on the day they left, you were trying to make friends by grilling up a huge batch of carrots and cabbage for everyone. Even though there was an involuntary *Gangnam Style* dance performance by all who ate at your stew buffet, your potential hobbit friends really over-reacted to the effect of the stew on their now barely functioning digestive systems. Soon after that, you were spotted stealing kidneys from the organ donor van as you wanted to spice up the stew.

Boy did that make everyone *really* mad.

So anyway, you've been tossed out onto the road and told never to return to the Shire. The whole thing's a daze, but you do now realize that a bit of paprika might have improved the taste. Hell's Kitchen at its finest.

It simplifies your life, since you can now pursue vengeance on Frodo, who, although only tangentially related to your life's failures, nonetheless deserves your full hatred. Think of yourself as Smeagol without a reason to be super-deranged. Really improves the ego, eh?

Job Details. Your task is to write C++ programs using MPI to help you pass the time while the Fellowship goes on a dumb and pointless quest. To do this, you're going to have to write many programs. They are described below.

1. **[10] Two Rings.** Deeply embedded in your cause of irritation is the fact that no one even *cared* that you had the Two Ring! In your opinion, the Two Ring is much better than the One Ring. Using math, it is possible to show that it is One Better, which is twice as good! Frodo argued that the One Ring is asymptotically the same as the Two Ring, and since he's cooler, he got to go with Gandalf. By the time you figured out what he was saying, it was cabbage time! (That'll teach you to skip algorithms class next time. . . .)

   You are to write a program as a tribute to the Two Ring, which is the best Ring in the World. Your program isn't the best Ring in the World, it's only a Tribute. (You have Tenaciously kept the ring, you see.) The first ring begins with process 0 and proceeds to send a message to the *next even* process, with the last process sending its message back to process 0. The second ring begins with process 1 and proceeds to send its message to the *previous odd* process, with the last process sending its message back to process 1. It does not matter which order the messages from different rings are sent in, as long as these are the only messages sent. Your code should work for any number of processes from 4 to 32. The message you send can be as simple as a number, or a more complicated message. Cruel points (very similar to cool points) can be earned for extremely cruel messages about Frodo.

For example, suppose $p = 10$ (the number of processes). The first ring would send a messages around the ring $0 \to 2 \to 4 \to 6 \to 8 \to 0$, and the second ring would send messages around $1 \to 9 \to 7 \to 5 \to 3 \to 1$.

2. **[10] Whack-An-Orc.** You followed the Fellowship through their many trials, as they defeated each encounter and grew wise in the ways of the world. Your preccccious (Two Ring) kept you hidden from both friend AND foe. (See, I told you it was twice as good.) It definitely wasn't the case that you were just perpetually overlooked, since you weren't important. Also, it would be difficult to accurately test the "friend" part of the hypothesis, since you have no friends. But we're not here to rub that in.

   Aaanyway, you eventually see Gimli and Legolas periodically yelling out the number of orcs they've killed in a given time period. They're having a grand old time, kind of the way friends do.

   Since we believe in magic, these numbers are conveniently contained within an array that is strewn all over the battlefield. They are computed using a pseudo-random number generator. Seed the generator with the number 71911 for this part of the project.

   You are to write a C++ program using MPI to find the maximum, minimum, and average value of a very large array of integer numbers. One might even say. . . one MILLION integer numbers. The array will be large enough that you will have to *scatter* the array into pieces to each process, have each process compute a partial answer, and then *reduce* back the answer together on process 0 after everyone is done. I would recommend that you do maximum completely, then do minimum, and then do the average. If you mix things up, your answers might be very weird, since messages can arrive out of order! And remember, you don't have to scatter the array each time, just once at the beginning!

3. **[10] In Your EYE.** Near the end of the journey, the huge eye in the sky started blinking uncontrollably. You realized, since no one else cared about this, that Sauron was trying to communicate using `MORdorSE` code. Either that, or a giant moth flew into his eye. Since you are trying to derive meaning out of useless things, you plan to do a frequency code analysis of the blinking to determine what Sauron is saying using `MORdorSE` code.

   You will write a parallel program that reads in a text file full of letters and figures out how many of each letter there are in his code. (The code will only use lowercase alphabetic letters.) After you do this, you'll probably want to run Sauron's eye through with a rusted tire iron. In his freaking eye.

   To parallelize this, you will need to have a process read the text file, *scatter* your book across the processes, and each process will have to count how many of each letter appears in that section of the code. Then you'll have to *gather* or *reduce* 26 counts back together. There is a bad way to write the local code, which is to loop over each letter of the alphabet and count how many of those there are and go from there. There is a better way that uses exactly one loop. Furthermore, this loop will not have a single conditional in it at all. Figure that out and write your code using this method.

Your project, in total, is worth 30 project points, and you will write each part in its own program. **This project must be done individually, and not in a group.** For each program, be sure to comment well, print out the program, and print out the output after running it. *Please staple your program together prior to turning it in.*

I'm around if you have any questions.