Ian Schweer
22514022
Part B:

Assuming the palindrome function is already there, this will be my guideline to implement part B.

```
Int main() {
        …… // code to initialize palindromes from file (TA provided)

        MPI_Init(argc, argv) // Initalize openMPI

        MPI_Comm_size(MPI_COMM_WORLD, &size) // get the size of the world
        MPI_Comm_rank(MPI_COMM_WORLD, &id) // get my position in the world

        PrepareBuffers() // prepare the buffer data with the palindromes to search through

        MPI_Scatter(data, datasize, MPI_CHAR,  recvbuf, recvsize, MPI_CHAR, id, comm);

        FindPalindrome() // Find my palindrom

        MPI_Gather(data, datasize, MPI_CHAR,  recvbuf, recvsize, MPI_CHAR, id, comm);

        MPI_Barrier // ensure everything gets here before stopping

        Find longest of long palindroms

        MPI_Exit() // close MPI

}
```

Part C:

Assuming I'm called with palindrome data

```
MyGather() {
        getLongestPalindrome()

        compare mydata with previous node (me – 1)

        send result to neighbor

}
```

Then at the root

```
MyGather_Root() {
```

```
        getLongestPalindrome()

        sendToNextNode()

        buffer r
         for each node n:
                r = MPI_recv on node n > |r| // take mpi_recv if greater then r

        findLongest() // longest should be first element
}
```