Questions

1. **How did you decide which type of socket to use? Why?**
   I decided to use the TCP socket because I was following the textbook. I also decided to use a TCP socket because TCP protocol is more reliable compared to UDP. In TCP, there is retransmission of lost data packets, which makes it more reliable and more suited for making http requests. I also saw that curl uses TCP when looking at the wireshark, which is what our program is trying to emulate.

2. **How did you choose the destination ports?**
   The destination ports were either the default port (80) if the user did not specify one in the command line arguments. If the user specified a destination port, the destination port would be the one the user specified in the command line argument. I got the user specified destination port through using regex to split the user inputted command line argument.

3. **What error handling cases did you implement?**
   - Check if url contains https by parsing through user input and checking
     - Reason: our program does not support https
   - Check if there is chunk encoding by parsing through http response and seeing if the transfer-encoding is chunked
     - Reason: our program is not required to support chunk encoding
   - TCP errors by checking by using the try and except. The try and except will catch the TCP errors
     - Reason: There might be HTTP request being issued to servers that don't exist or when port 443 is set as the destination port

4. **How does your program terminate? What happens to the TCP connection?**
   My program terminates after running through all the code in my program. My program should terminate after logging an entry into the Log.csv. My program could terminate early if it's part of my error checking. The TCP connection is closed before my program terminates through closing the sockets.

5. **For the unsuccessful URLs, why were they unsuccessful?**
   One of the reasons the unsuccessful URLs were unsuccessful is because some of the requested objects are returned with chunk encoding, which our program does not support. Thus making it unsuccessful. Another reason is because the URL is malformed. This means the url does not comply with HTTP standards. An example of this is "http://wwwexample.com/index.html", one of the given websites for testing. The "wwwexample" part does not comply with HTTP standards. Another reason is that the HTTP request was being sent to a server that does not exist. Another reason is the HTTP request is being sent to 443, which causes errors and makes the URL unsuccessful. Port 443 is the default port for using HTTPS, and you cannot serve both HTTP and HTTPS on the same port.

6. **What happens if you try to access a site using HTTPS?**

If you try to access a site using HTTPS, there will be an error message printed and the program will exit. This is because our program does not support HTTPS.