My *first test* was focused on the basic functionality of my program. I would simply input the command provided in the documentation: *./myweb www.example.com 93.184.216.34:80/index.html*. It is expected for the response body to be written to the *output.dat* file found in the *bin* directory. There should be no errors during this process and the program should terminate gracefully after completion. After running the aforementioned command, my program did exactly as expected.

The *second test* was now testing the robustness of my error handling in improper *Ip addressing.* For instance, using the same command as in *Test 1*, however changing the *Ip address* from *93.184.216.34* to something like *89.185.216.34*, it is expected for the program to terminate after failing to reach the requested server **after 60 seconds**. Running the following command for my code: *./myweb neverssl.com 89.185.216.34/index.html*, the *client* waits to connect to the server but times out and returns an error message: *Failed to Connect to Server*. My program then gracefully terminates.

The *thirst test* will monitor the response of the server when running into a *400 error.* To test for this we run the provided website *Ip address*: *34.223.124.45*. This *Ip address* will be used with the following command: *./myweb www.example.com 34.223.124.45/index.html -h.* The expected output should return a *status code 401 unauthorized permission error*. Additionally, because the *-h* flag was set, the server response should be outputted to *cout* and not be written in *output.dat*. The actual output follows this exact expectation, and the status code error is outputted to the command line.

The *fourth test* will test our byte buffering and ensure that every byte is read and processed by the program. This will be done through a change in our *Macros* instead of a specific command line execution. To do this I changed the *BUFFER_SIZE Macro* to be *10* (was *2046*) then re-ran the original command in *Test 1*. The reason for this was to ensure that the passing of bytes from the buffer array (which takes the bytes from the server response) is properly passing them onward towards the output vector. This is done to ensure that response sizes are being read chunk by chunk instead of whatever last populated the buffer. Obviously, the expected return is for the full response body. After running the mentioned command, after changing the file *Macro*, the *output.dat* file contains the full response body.

The *fifth test* will ensure that *regex* parsing is done correctly and efficiently. To do this we test for a number of possible input commands that vary in domain name types. These can range from *https:// , http:// , www. , .com, .com.uk , etc.* This is to ensure that requests are accurate and responses are to our expectations. To do this I ran a series of commands issuing these domain names: *http://www.softwareqatest.com, http://www.worldslongestwebsite.com, http://pudim.com.br*. Additionally, I had also made sure to test for scenarios where the file path was not specified. In this case a *200* response was expected.