General structure explanation:

When the browser client is going to make a request for the first time. The program tries to open a cache file named with url (filename to be "www.google.com" for google site), and raises an IO Exception because it does not exist. Then, the proxy needs to forward requests to the remote server and retrieve the webpage. The proxy will then receive a file from the server, cache it to tmpFile, and forward it to the client.

The next time the client requests the same file, instead of having to query to remote server, your proxy will simply read the appropriate webpage from its cache file in the local directory and send it directly back to the client.

Note that the proxy server keeps a **server socket** the whole time to receive connection from browser client and send response to it. But if there is no cache, it needs to create a new **client socket** to connect to the remote server and retrieve the webpage.

A few tips:

a. After sending request, it's better to wait a few milliseconds before recv to make sure (time.wait() can be helpful)
b. select is useful to check if there is any more to read from socket.
c. Instead of using makefile in the skeleton code, it's easier to do socket.send() and socket.recv() directly. Make sure to encode() string when sending and decode() what you received
d. For debugging, it's helpful to print down what you send to socket and receive from socket.
e. Useful commands to check ports availability: lsof -i -n -P | grep <port number>

Recommendation for development process:

1. Make sure to get the request message from your browser.
2. Create a socket on the proxyserver and send requests to the website, and make sure you get the responses from the website. (check by looking at the printing out messages and also the file saved in the current directory)
3. Finish the block of read from the local directory file and forward to the client.
4. Finish the HTTP response message for files not found and remember to close the sockets.