Please answer the following questions for the project report.

1. Team details: Clearly state the names and netids of your team members (there are 2 of you).
    a. Arushi Pradhan (ap2132) and Shahrez Ahmad(ssa136)
2. Collaboration: Who did you collaborate with on this project? What resources and references did you consult? Please also specify on what aspect of the project you collaborated or consulted.
    a. To understand the Project's instructions, we consulted the professor and one of the TAs about some of the specifics with the project such as whether to follow TCP or UDP and about how the socket functionality would work with 3 servers and a client. We also consulted the internet and sites such as Stack Overflow and GeeksforGeeks when it came to looking up Python syntax. Initially, Arushi worked on the client.py and rs.py files while Shahrez worked on the ts1.py and ts2.py files. Afterwards, we collaborated over zoom to test, fix bugs and add additional implementation.
3. Briefly discuss how you implemented recursive and iterative query functionality. Please be clear and specific.
    a. According to the instructions, if the client sends a query with an "it"(iterative) flag, the root server checks its dictionary and returns the query with a "ns"(name server) flag and the hostname of the corresponding top-level server instead. On the client side, the code uses an if statement to check for a "ns" flag, sends a new query to the top-level server and saves the response to the resolved.txt file. Else, if there isn't an "it" flag, the code assumes it to be "rd" (recursion desired) and adds the response to the resolved.txt file.
4. Is there any portion of your code that does not work as required above? Please explain.
    a. After hours of debugging and collaboration, our code works exactly as intended :)
5. Did you encounter any difficulties? If so, explain.
    a. We needed clarification on socket ports, specifically on why listening ports are needed on the server side when a client initiates a connection. We also encountered an issue when it came to incrementing the identification key. This issue was related to how Python processes variables in format strings. Another issue we had with Python is that even if we can check for an object's emptiness using "If object==True", this wouldn't apply for lists. Instead, we had to use a "not (not list)" statement to determine if a list was not empty.

6. What did you learn from working on this project? Add any interesting observations not otherwise covered in the questions above. Please be specific and technical in your response.
    a. We learned that Python, as a language, has an issue with processing variables in format strings. Originally, we incremented the value of the identification key in the format string, but we later found out that this change was not reflected in the output. After tracing the code, we manually incremented the variable before adding it to the string, which produced the desired result. Something interesting that we also observed was how persistent and non persistent connections functioned in code. In class, we just learn about their behaviors but we only fully realized its meaning when we had to figure out when to close and open the sockets and how to go about doing so. For instance, Arushi went with a non persistent approach in her files while Shahrez went with a persistent approach in his files. In Arushi's case, she would open and close a connection every time a query was sent out and a response was received, while Shahrez just kept his open and closed it at the very end. Eventually, when we collaborated, we changed some of Shahrez's code to match with Arushi's non persistent approach. Lastly, we were able to take advantage of the non-persistent approach to open a connection and send a final "done" message from the client to the root server as a request to shut down all the servers.