README (Instructions to execute)

Q1) Basic Folder

First run the server.py file and then the client.py file as given below.

Client.py: Requests user to select the type of request and enter the appropriate values

INPUT:

- It asks for the IP address of the server(10.0.1.2 in this case)

OUTPUT:

- Response is received from the server with the appropriate code like 200, 404.

Server.py: This file receives requests from client and sends the packets also with the appropriate codes. The data at the server is stored in the local memory(a python dictionary here). The server is initially empty. So, you have to populate the dictionary with 6 put requests(or more if you want or any delete requests if you need). So, now you can get all the keys from the server.

INPUT:

- Enter the IP address: 10.0.1.2 (IP address of the server taken to be 10.0.1.2)

OUTPUT:

- The file sends required packets to the client which you can see on the wireshark with the appropriate codes.(Ex. HTTP =/1.1 200 OK Value)

Q2) WebCache Development(Star Folder)

Here, you should run the server.py first, then, cache.py and then client.py .

Client.py: This file sends get requests from the client first to the cache.

INPUT :

- It asks the user for the IP address of the cache(10.0.1.2 in this case)
- Though, this also asks for the type of request that you want to send, you can only ask for the GET requests(which was asked in the question)

Cache.py: This file checks if the packet requested by the client is there in the cache itself. If it is not, then it asks the server for that packet and then sends the packet to the client. If the packet is not there even in the server.py, then it sends the appropriate response that it is not there. Every packer request that is sent through the cache is stored in the cache(here a python dictionary is used for storing).

INPUT:

- This asks for the IP address of cache(10.0.1.2) .

OUTPUT:

- The packets are sent to the client and server and received from the client which can be seen on the wireshark.

Server.py: This receives requests from the cache and sends appropriate responses back to the client. This file als uses a dictionary to store the data.

INPUT:

- Asks the user for the IP address of the server(10.0.1.3)

The requests will be in the following form for each request:

PUT: PUT /A/key/value HTTP/1.1

GET: GET /A?key=key HTTP/1.1
DELETE: DELETE /A/key HTTP/1.1

PCAP FILES:

All the initial get, put, delete requests are done in a same pcap file and you can observe the packet transfer at h1, h2 and h1, h2, h3(for web cache)
There is also a get_requests pcap file for both basic and star that has all the get requests using which I have calculated the time.