Exercise 3: Using Wireshark to understand basic HTTP request/response messages (marked, include in your report)

Question 1: What is the status code and phrase returned from the server to the client browser?

Status code: 200

Phrase returned: OK

Question 2: When was the HTML file that the browser is retrieving last modified at the server? Does the response also contain a DATE header? How are these two fields different?

The HTML file was last modified at Tue, 23 Sep 2003 05:29:00 GMT.

Yes, the response also contains a DATE header.

The difference is that the DATE header specifies the time when server produce and send the message, whereas the last modified time is the time when the objects have been made or changed.

Question 3: Is the connection established between the browser and the server persistent or non-persistent? How can you infer this?

The connection established is persistent, and this can be seen from the Connection header:

Connection: Keep-Alive

Question 4: How many bytes of content are being returned to the browser?

73 Bytes of content are being returned.

Accept-Ranges: bytes

Content-Length: 73

Question 5: What is the data contained inside the HTTP response packet?

The response packet includes a html file: lab2-1.html

Exercise 4: Using Wireshark to understand the HTTP CONDITIONAL GET/response interaction (marked, include in your report)

Question 1: Inspect the contents of the first HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?
No.

Question 2: Does the response indicate the last time that the requested file was modified?
Yes, it indicates that the HTML file was last modified at Tue, 23 Sep 2003 05:35:00 GMT.

Question 3: Now inspect the contents of the second HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE:" and "IF-NONE-MATCH" lines in the HTTP GET? If so, what information is contained in these header lines?
Yes, it contains 'IF-MODIFIED-SINCE: "and 'IF-NONE-MATCH. The Conditional GET contains the request for checking whether the object have been changed since a specific date.

Question 4: What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.
Status code: 304
Phrase returned: Not Modified
The server does not explicitly return the contents of the file, because the file remains unchanged.

Question 5: What is the value of the Etag field in the 2nd response message and how it is used? Has this value changed since the 1 st response message was received?
2nd response: ETag: "1bfef-173-8f4ae900"
1st response: ETag: "1bfef-173-8f4ae900"
Both ETags are the same and this is how it works:
After we request the first HTTP GET request, server returns the responding objects and its ETag values. Later, when we request the second HTTP GET request, the message which contains the saved ETag value and a segment "If-None-Math: "ETag-value" got sent to the server, and the server will compare received ETag value with the ETag value of the object, if they match, then returns a short response which contains 304 Not Modified (Our case). However, if they don't match, a full response message will be returned which contains the object content.

(*) Exercise 5: Ping Client

```python
from socket import *
import time,sys
from statistics import mean
clientSocket=socket(AF_INET,SOCK_DGRAM)
host=sys.argv[1]
port_number=int(sys.argv[2])
i=0
rtt_list=[]
lost=0
while i<10:
    localtime = time.asctime( time.localtime(time.time()) )
    message=str.encode('Ping'+' '+str(i)+' '+str(localtime))
    try:
        start_time=time.time()
        clientSocket.sendto(message,(host,port_number))
        clientSocket.settimeout(1.0)
        message1,address=clientSocket.recvfrom(1024)
        end_time=time.time()
        rtt=(end_time-start_time)*1000
        rtt_list.append(rtt)
        print(f'ping to {host}, seq = {i}, rtt = {rtt} ms')
        time.sleep(1)
        i+=1
    except:
        print(f'ping to {host}, seq = {i}, time out')
        lost+=1
        i+=1
if lost==10:
        print('All request time out')
else:
        min_rtt = min(rtt_list)
        max_rtt = max(rtt_list)
        aver_rtt = mean(rtt_list)
        lost_rate=(lost/10)*100
        print(f'10 packets transmitted, {10-lost} packets received,{lost_rate}% packet loss')
        print(f'round-trip time min/avg/max = {min_rtt} ms,{aver_rtt} ms,{max_rtt} ms')
```