Problem 1

Suppose you have a new computer just set up. `dig` is one of the most useful DNS lookup tool. You can check out the manual of `dig` at `http://linux.die.net/man/1/dig`. A typical invocation of `dig` looks like: `dig @server name type`.

Suppose that on April 19, 2017 at 15:35:21, you have issued "`dig google.com a`" to get an IPv4 address for `google.com` domain from your caching resolver and got the following result:

```
; <<>> DiG 9.8.3-P1 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17779
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             239     IN      A       172.217.4.142

;; AUTHORITY SECTION:
google.com.             55414   IN      NS      ns4.google.com.
google.com.             55414   IN      NS      ns2.google.com.
google.com.             55414   IN      NS      ns1.google.com.
google.com.             55414   IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         145521  IN      A       216.239.32.10
ns2.google.com.         215983  IN      A       216.239.34.10
ns3.google.com.         215983  IN      A       216.239.36.10
ns4.google.com.         215983  IN      A       216.239.38.10

;; Query time: 81 msec
;; SERVER: 128.97.128.1#53(128.97.128.1)
;; WHEN: Wed Apr 19 15:35:21 2017
;; MSG SIZE rcvd: 180
```

1. What is the discovered IPv4 address of `google.com` domain?
2. If you issue the same command 1 minute later, how would "ANSWER SECTION" look like?
3. When would be the earliest (absolute) time the caching resolver would contact one of the `google.com` name servers again?
4. When would be the earliest (absolute) time the caching resolver would contact one of the `.com` name servers?

Problem 1 continued on next page...

1. 172.217.4.142
2. google.com. 179 IN A 172.217.4.142
3. Wed Apr 19 15:39:20 2017 (Wed Apr 19 15:35:21 2017 + 239 sec)
4. Thu Apr 20 06:58:55 2017 (Wed Apr 19 15:35:21 2017 + 55414 sec)

Problem 2

Suppose that you walked into Boelter Hall and get connected to `CSD` WiFi network, which automatically gave you IP address of the local caching resolver. However, initially, it doesn't allow you to do anything unless you type your username and password in a popup window (or if you try to go to any website in your browser).

1. Explain a mechanism of how does the "`CSD`" network achieve this / which features of DNS/HTTP make it possible.

The captive portal can achieve this.
The DNS hijacking redirects unauthenticated clients to the authentication page. That is, the local caching DNS server will return the IP address of the captive portal page as a result of all DNS lookups.
The HTTP server authenticates the client with the username/password provided, and issues HTTP redirect code (e.g. 302 Found) to automatically redirect the client to their requested web page.
Note: redirection code is optional.

Problem 3

Same context as Problem 2. After you successfully logged in, you can start using the Internet. Suppose the caching resolver has just rebooted and its cache is completely empty; RTT between your computer and the caching resolver is $10ms$ and RTT between the caching resolver and any authoritative name server is $100ms$; all responses have TTL 12 hours.

1. If you try to go to `ucla.edu`, what would be minimum amount of time you will need to wait before your web browser will be able to initiate connect to the UCLA's web server?
2. What would be the time, if a minute later you will decide to go to `ccle.ucla.edu`?
3. What would be the time, if another minute later you will decide to go to `piazza.com`?
4. What would be the time, if another minute later you will decide to go to `gradescope.com`?

1. 310 ms
2. 110 ms. (or 210 if ccle.ucla.edu happen to be delegated to a physically different name server)
3. 310 ms. (or 210 if .com nameserver happen to know A record for piazza.com. Doest usually happen.
4. 210 ms. (or 110 for the same reason as in 3)

Problem 4

How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain.

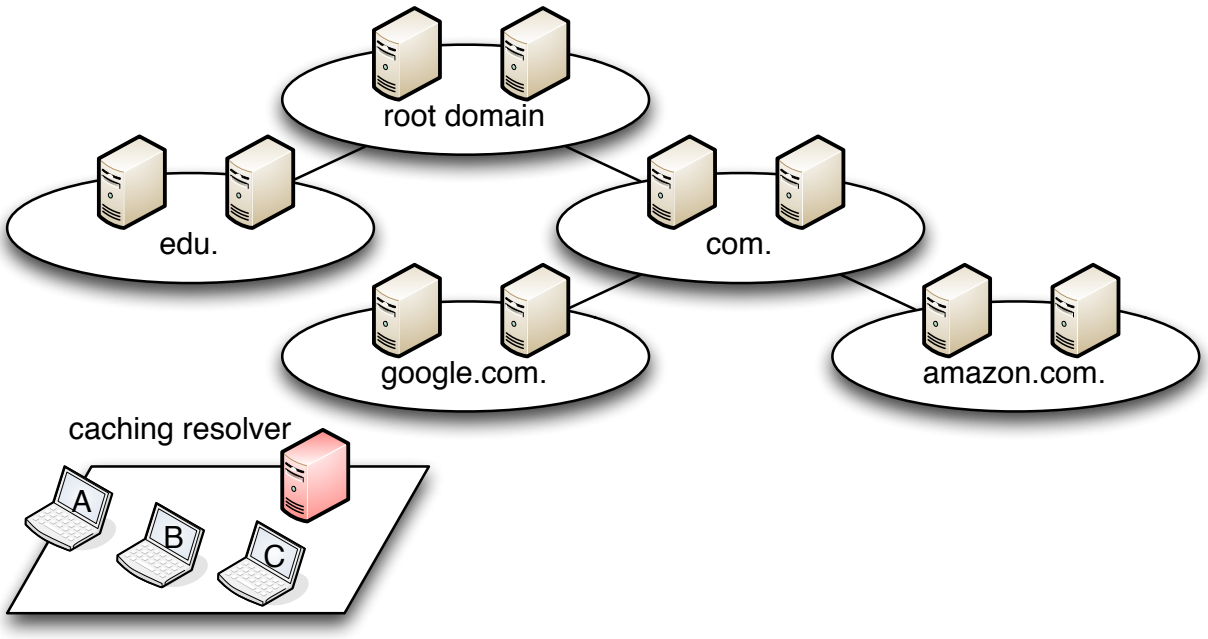
SMTP uses a line containing only a period to mark the end of a message body. HTTP uses Content-Length header field to indicate the length of a message body. No, HTTP cannot use the method used by SMTP, because HTTP message could be binary data, whereas in SMTP, the message body must be in 7-bit ASCII format.

Problem 5

Consider the following environment with a local DNS caching resolver and a set of authoritative DNS name servers.

Assume that initially,

- the caching resolver cache is empty,
- TTL values for all records is 1 hour,
- RTT between stub resolvers (hosts A, B, and C) and the caching resolver is 20 ms,
- RTT between the caching resolver and any of the authoritative name servers is 150 ms,
- There are no packet losses,
- All processing delays are 0 ms



1. At T=0 min, Host-A sends a query for A record for amazon.com, and after receiving the answer sends a query for A record for www.amazon.com. How long did it take to receive all the answers?
2. At T=40 min, Host-B sends a query for MX record for google.com that returns

```
google.com.           3600    IN      MX      10 primary.google.com.
google.com.           3600    IN      MX      30 backup.google.com.
primary.google.com.   3600    IN      A       74.125.28.27
backup.google.com.    3600    IN      A       173.194.211.27
```

   (Similar to NS records, the DNS server may return glue A/AAAA records in addition to the requested MX records.) How long did it take to get the answer?
3. At T=70 min, Host-C sends a query for AAAA (IPv6) record for mail.google.com, following at T=75 mins with a query for AAAA (IPv6) record for hangout.google.com. How long did it take for Host-C to receive each of the answers (i.e., relative to T=70min for the first, and relative to T=75 mins for the second)?
4. List DNS records that the caching resolver has at T=90 minutes

Problem 5 continued on next page...

NOTE: Need to confirm with midterm answer.

1. 470ms for amazon.com : 20 + 150*3 ms (root, .com, amazon.com)
   170ms for www.amazon.com : 20 + 150 (amazon.com)
   470 + 170 = 640 ms

2. 320 ms : 20 + 150*2 ms (already have ip of domain name server for .com)

3. 170ms for mail.google.com : 20 + 150 ms
   170ms for hangout.google.com : 20 + 150 ms
   (already have ip of domain name server for google.com, need AAAA record)

4. google.com/MX
   primary.google.com/A
   backup.google.com/A
   google.com/NS
   mail.google.com/AAAA
   hangout.google.com/AAAA
   Note: Just answer all A/AAAA score can get full scores this time.

   There should also be the A record containing the IP address of the google.com NS. But we did not give out the name of the google.com NS, so you do not have to write it to get full mark.