Exercise 3: Digging into DNS (marked, include in the lab report)

In order to answer the following questions, you will make DNS queries using some of the query types you have encountered in the above exercise. Some questions require you to make multiple DNS queries. Before you proceed, read the manpage of dig (type man dig in the terminal). Make sure you understand how you can explicitly specify the following:

- nameserver to query
- type of DNS query to make (the default query types are those you saw in exercise 1)
- performing reverse queries

Note: Include the output of all the dig commands you have used in your answers.

To send a query to a particular name server (say x.x.x.x) you should use the following command:

```
dig @x.x.x.x hostname
```

Question 1. What is the IP address of www.cecs.anu.edu.au . What type of DNS query is sent to get this answer?

Answer 1: The IP address is 150.203.161.98, the type of DNS query is A.

```
z5195349@vx3:/tmp_amd/reed/export/reed/2/z5195349$ dig www.cecs.anu.edu.au

; <<>> DiG 9.7.3 <<>> www.cecs.anu.edu.au
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22347
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 6

;; QUESTION SECTION:
;www.cecs.anu.edu.au.            IN      A

;; ANSWER SECTION:
www.cecs.anu.edu.au.    3600    IN      CNAME   rproxy.cecs.anu.edu.au.
rproxy.cecs.anu.edu.au. 3600    IN      A       150.203.161.98

;; AUTHORITY SECTION:
cecs.anu.edu.au.        1800    IN      NS      ns3.cecs.anu.edu.au.
cecs.anu.edu.au.        1800    IN      NS      ns2.cecs.anu.edu.au.
cecs.anu.edu.au.        1800    IN      NS      ns4.cecs.anu.edu.au.

;; ADDITIONAL SECTION:
ns2.cecs.anu.edu.au.    1193    IN      A       150.203.161.36
ns2.cecs.anu.edu.au.    1800    IN      AAAA    2001:388:1034:2905::24
ns3.cecs.anu.edu.au.    2427    IN      A       150.203.161.50
ns3.cecs.anu.edu.au.    1800    IN      AAAA    2001:388:1034:2905::32
ns4.cecs.anu.edu.au.    1193    IN      A       150.203.161.38
ns4.cecs.anu.edu.au.    1800    IN      AAAA    2001:388:1034:2905::26

;; Query time: 75 msec
;; SERVER: 129.94.242.45#53(129.94.242.45)
;; WHEN: Wed Aug 15 09:48:51 2018
;; MSG SIZE  rcvd: 260
```

Question 2. What is the canonical name for the CECS ANU web server? What is its IP address? Suggest a reason for having an alias for this server.

Answer 2: Observe the output above that the canonical name is rproxy.cecs.anu.edu.au. Its IP address is 150.203.161.98. Reason: sometimes a server may change its IP address, if we use 'A' record to record, we have to change our record when server's IP address changes. However, 'CNAME' record just records domain name, so we don't need to change our IP address when server's IP address changes.

```
;; ANSWER SECTION:
www.cecs.anu.edu.au.       274      IN      CNAME   rproxy.cecs.anu.edu.au.
rproxy.cecs.anu.edu.au. 274        IN      A       150.203.161.98
```

Question 3. What can you make of the rest of the response (i.e. the details available in the Authority and Additional sections)?

Answer 3: The Authority sections show that some authority nameservers can be used (also means some resource record in type 'NS'), and the Additional sections show the IP address ('A' is IPv4, and 'AAAA' is IPv6) of authority nameservers which above section shows. Also, in the end of the response, it shows other detail, like Query time (the time of executing this query), local server's IP address, date and time.

```
;; AUTHORITY SECTION:
cecs.anu.edu.au.           1800     IN      NS      ns3.cecs.anu.edu.au.
cecs.anu.edu.au.           1800     IN      NS      ns2.cecs.anu.edu.au.
cecs.anu.edu.au.           1800     IN      NS      ns4.cecs.anu.edu.au.

;; ADDITIONAL SECTION:
ns2.cecs.anu.edu.au.       1193     IN      A       150.203.161.36
ns2.cecs.anu.edu.au.       1800     IN      AAAA    2001:388:1034:2905::24
ns3.cecs.anu.edu.au.       2427     IN      A       150.203.161.50
ns3.cecs.anu.edu.au.       1800     IN      AAAA    2001:388:1034:2905::32
ns4.cecs.anu.edu.au.       1193     IN      A       150.203.161.38
ns4.cecs.anu.edu.au.       1800     IN      AAAA    2001:388:1034:2905::26

;; Query time: 75 msec
;; SERVER: 129.94.242.45#53(129.94.242.45)
;; WHEN: Wed Aug 15 09:48:51 2018
;; MSG SIZE  rcvd: 260
```

Question 4. What is the IP address of the local nameserver for your machine?

Answer 4: Local nameserver's IP address is 129.94.242.45

```
;; Query time: 75 msec
;; SERVER: 129.94.242.45#53(129.94.242.45)
;; WHEN: Wed Aug 15 09:48:51 2018
;; MSG SIZE  rcvd: 260
```

Question 5. What are the DNS nameservers for the "cecs.anu.edu.au" domain (note: the domain name is cecs.anu.edu.au and not www.cecs.anu.edu.au)? Find out their IP addresses? What type of DNS query is sent to obtain this information?

Answer 5: The DNS nameservers are ns2.cecs.anu.edu.au, ns3.cecs.anu.edu.au, ns4.cecs.anu.edu.au, the IP addresses are 150.203.161.36(IPv4)/2001:388:1034:2905::24(IPv6), 150.203.161.50(IPv4)/2001:388:1034:2905::32(IPv6),150.203.161.38(IPv4)/2001:388:1034:2905::26(IPv6) respectively. The type is NS which means delegating a DNS zone to use the given authoritative name servers

```
z5195349@vx3:/tmp_amd/reed/export/reed/2/z5195349$ dig cecs.anu.edu.au NS

; <<>> DiG 9.7.3 <<>> cecs.anu.edu.au NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1656
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 6

;; QUESTION SECTION:
;cecs.anu.edu.au.                IN      NS

;; ANSWER SECTION:
cecs.anu.edu.au.        545     IN      NS      ns2.cecs.anu.edu.au.
cecs.anu.edu.au.        545     IN      NS      ns4.cecs.anu.edu.au.
cecs.anu.edu.au.        545     IN      NS      ns3.cecs.anu.edu.au.

;; ADDITIONAL SECTION:
ns2.cecs.anu.edu.au.    719     IN      A       150.203.161.36
ns2.cecs.anu.edu.au.    545     IN      AAAA    2001:388:1034:2905::24
ns3.cecs.anu.edu.au.    719     IN      A       150.203.161.50
ns3.cecs.anu.edu.au.    545     IN      AAAA    2001:388:1034:2905::32
ns4.cecs.anu.edu.au.    719     IN      A       150.203.161.38
ns4.cecs.anu.edu.au.    545     IN      AAAA    2001:388:1034:2905::26

;; Query time: 1 msec
;; SERVER: 129.94.242.45#53(129.94.242.45)
;; WHEN: Wed Aug 15 09:56:45 2018
;; MSG SIZE  rcvd: 219
```

Question 6. What is the DNS name associated with the IP address 149.171.158.109?
What type of DNS query is sent to obtain this information?

**Answer 6: For this, we use reverse DNS. When we observe the answer section, we
can find that it has www.engineering.unsw.edu.au which suggests that this
machine hosts the website. Also, we can find the engplws008.ad.unsw.edu.au and
engplws008.eng.unsw.edu.au which mean that the same server may be hosting
some active directory services. The type is PTR (the most common use is for
implementing reverse DNS lookups)**

```
z5195349@vx3:/tmp_amd/reed/export/reed/2/z5195349$ dig -x 149.171.158.109

; <<>> DiG 9.7.3 <<>> -x 149.171.158.109
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18797
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 3, ADDITIONAL: 4

;; QUESTION SECTION:
;109.158.171.149.in-addr.arpa.  IN      PTR

;; ANSWER SECTION:
109.158.171.149.in-addr.arpa. 1926 IN   PTR     engplws008.ad.unsw.edu.au.
109.158.171.149.in-addr.arpa. 1926 IN   PTR     engplws008.eng.unsw.edu.au.
109.158.171.149.in-addr.arpa. 1926 IN   PTR     www.engineering.unsw.edu.au.

;; AUTHORITY SECTION:
158.171.149.in-addr.arpa. 526   IN      NS      ns2.unsw.edu.au.
158.171.149.in-addr.arpa. 526   IN      NS      ns1.unsw.edu.au.
158.171.149.in-addr.arpa. 526   IN      NS      ns3.unsw.edu.au.

;; ADDITIONAL SECTION:
ns1.unsw.edu.au.        3982    IN      A       129.94.0.192
ns2.unsw.edu.au.        3982    IN      A       129.94.0.193
ns3.unsw.edu.au.        3982    IN      A       192.155.82.178
ns3.unsw.edu.au.        3294    IN      AAAA    2600:3c01::f03c:91ff:fe73:5f10

;; Query time: 1 msec
;; SERVER: 129.94.242.45#53(129.94.242.45)
;; WHEN: Wed Aug 15 10:00:16 2018
;; MSG SIZE  rcvd: 274
```

Question 7. Run dig and query the CSE nameserver (129.94.242.33) for the mail servers
for Yahoo! Mail (again the domain name is yahoo.com, not www.yahoo.com). Did you
get an authoritative answer? Why? (HINT: Just because a response contains information
in the authoritative part of the DNS response message does not mean it came from an
authoritative name server. You should examine the flags in the response to determine
the answer)

**Answer 7: No, we didn't get an authoritative answer because the flags do not
contain 'AA', which means authoritative answer. This is because it has authority
for only this domain we are in but not for the Yahoo domain.**

```
z5195349@vx3:/tmp_amd/reed/export/reed/2/z5195349$ dig @129.94.242.33 yahoo.com MX

; <<>> DiG 9.7.3 <<>> @129.94.242.33 yahoo.com MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10958
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 8

;; QUESTION SECTION:
;yahoo.com.                     IN      MX

;; ANSWER SECTION:
yahoo.com.              1800    IN      MX      1 mta6.am0.yahoodns.net.
yahoo.com.              1800    IN      MX      1 mta7.am0.yahoodns.net.
yahoo.com.              1800    IN      MX      1 mta5.am0.yahoodns.net.

;; AUTHORITY SECTION:
yahoo.com.              18126   IN      NS      ns1.yahoo.com.
yahoo.com.              18126   IN      NS      ns3.yahoo.com.
yahoo.com.              18126   IN      NS      ns4.yahoo.com.
yahoo.com.              18126   IN      NS      ns2.yahoo.com.
yahoo.com.              18126   IN      NS      ns5.yahoo.com.

;; ADDITIONAL SECTION:
ns1.yahoo.com.          192960  IN      A       68.180.131.16
ns1.yahoo.com.          100355  IN      AAAA    2001:4998:130::1001
ns2.yahoo.com.          175842  IN      A       68.142.255.16
ns2.yahoo.com.          78896   IN      AAAA    2001:4998:140::1002
ns3.yahoo.com.          366826  IN      A       203.84.221.53
ns3.yahoo.com.          8650    IN      AAAA    2406:8600:b8:fe03::1003
ns4.yahoo.com.          182646  IN      A       98.138.11.157
ns5.yahoo.com.          168717  IN      A       119.160.253.83

;; Query time: 163 msec
;; SERVER: 129.94.242.33#53(129.94.242.33)
;; WHEN: Wed Aug 15 10:10:49 2018
;; MSG SIZE  rcvd: 360
```

Question 8. Repeat the above (i.e. Question 7) but use one of the nameservers obtained in Question 5. What is the result?

Answer 8: We can find the status is REFUSED, so we do not get a response. The reason could be to prevent some DNS attacks, these servers do not respond every query from Internet, they just respond some certain IP addresses (The network in ANU).

```
z5195349@vx3:/tmp_amd/reed/export/reed/2/z5195349$ dig @150.203.161.36 yahoo.com MX

; <<>> DiG 9.7.3 <<>> @150.203.161.36 yahoo.com MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 48201
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;yahoo.com.                     IN      MX

;; Query time: 7 msec
;; SERVER: 150.203.161.36#53(150.203.161.36)
;; WHEN: Wed Aug 15 10:18:38 2018
;; MSG SIZE  rcvd: 27
```

Question 9. Obtain the authoritative answer for the mail servers for Yahoo! mail. What type of DNS query is sent to obtain this information?

Answer 9: We can use nameservers from yahoo.com in Q7. One of nameservers is ns1.yahoo.com, IP address is 68.180.131.16. The type is MX.

```
z5195349@vx1:/tmp_amd/reed/export/reed/2/z5195349$ dig @68.180.131.16 yahoo.com MX

; <<>> DiG 9.7.3 <<>> @68.180.131.16 yahoo.com MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22681
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 8
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;yahoo.com.                     IN      MX

;; ANSWER SECTION:
yahoo.com.              1800    IN      MX      1 mta6.am0.yahoodns.net.
yahoo.com.              1800    IN      MX      1 mta7.am0.yahoodns.net.
yahoo.com.              1800    IN      MX      1 mta5.am0.yahoodns.net.

;; AUTHORITY SECTION:
yahoo.com.              172800  IN      NS      ns4.yahoo.com.
yahoo.com.              172800  IN      NS      ns5.yahoo.com.
yahoo.com.              172800  IN      NS      ns3.yahoo.com.
yahoo.com.              172800  IN      NS      ns1.yahoo.com.
yahoo.com.              172800  IN      NS      ns2.yahoo.com.

;; ADDITIONAL SECTION:
ns1.yahoo.com.          1209600 IN      A       68.180.131.16
ns2.yahoo.com.          1209600 IN      A       68.142.255.16
ns3.yahoo.com.          1209600 IN      A       203.84.221.53
ns4.yahoo.com.          1209600 IN      A       98.138.11.157
ns5.yahoo.com.          1209600 IN      A       119.160.253.83
ns1.yahoo.com.          86400   IN      AAAA    2001:4998:130::1001
ns2.yahoo.com.          86400   IN      AAAA    2001:4998:140::1002
ns3.yahoo.com.          86400   IN      AAAA    2406:8600:b8:fe03::1003

;; Query time: 145 msec
;; SERVER: 68.180.131.16#53(68.180.131.16)
;; WHEN: Wed Aug 15 14:47:42 2018
;; MSG SIZE  rcvd: 360
```

Question 10. In this exercise you simulate the iterative DNS query process to find the IP address of your machine (e.g. lyre00.cse.unsw.edu.au). First, find the name server (query type NS) of the "." domain (root domain). Query this nameserver to find the authoritative name server for the "au." domain. Query this second server to find the authoritative nameserver for the "edu.au." domain. Now query this nameserver to find the authoritative nameserver for "unsw.edu.au". Next query the nameserver of unsw.edu.au to find the authoritative name server of cse.unsw.edu.au. Now query the nameserver of cse.unsw.edu.au to find the IP address of your host. How many DNS servers do you have to query to get the authoritative answer?

Answer 10: First, find root domain's nameservers:

```
z5195349@vx3:/tmp_amd/reed/export/reed/2/z5195349$ dig . NS

; <<>> DiG 9.7.3 <<>> . NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12326
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13

;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                      31375    IN      NS      l.root-servers.net.
.                      31375    IN      NS      m.root-servers.net.
.                      31375    IN      NS      a.root-servers.net.
.                      31375    IN      NS      h.root-servers.net.
.                      31375    IN      NS      i.root-servers.net.
.                      31375    IN      NS      g.root-servers.net.
.                      31375    IN      NS      b.root-servers.net.
.                      31375    IN      NS      e.root-servers.net.
.                      31375    IN      NS      j.root-servers.net.
.                      31375    IN      NS      k.root-servers.net.
.                      31375    IN      NS      f.root-servers.net.
.                      31375    IN      NS      d.root-servers.net.
.                      31375    IN      NS      c.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net.    188562   IN      A       198.41.0.4
a.root-servers.net.    154171   IN      AAAA    2001:503:ba3e::2:30
b.root-servers.net.    18879    IN      A       199.9.14.201
b.root-servers.net.    385100   IN      AAAA    2001:500:200::b
c.root-servers.net.    3211     IN      A       192.33.4.12
c.root-servers.net.    13038    IN      AAAA    2001:500:2::c
d.root-servers.net.    434737   IN      A       199.7.91.13
d.root-servers.net.    434737   IN      AAAA    2001:500:2d::d
e.root-servers.net.    21248    IN      A       192.203.230.10
e.root-servers.net.    396000   IN      AAAA    2001:500:a8::e
f.root-servers.net.    274783   IN      A       192.5.5.241
f.root-servers.net.    60200    IN      AAAA    2001:500:2f::f
g.root-servers.net.    21248    IN      A       192.112.36.4

;; Query time: 0 msec
;; SERVER: 129.94.242.45#53(129.94.242.45)
;; WHEN: Wed Aug 15 10:47:12 2018
;; MSG SIZE  rcvd: 508
```

Secondly, query 'au.' domain:

```
z5195349@vx1:/tmp_amd/reed/export/reed/2/z5195349$ dig @198.41.0.4 lyre00.cse.unsw.edu.au NS

; <<>> DiG 9.7.3 <<>> @198.41.0.4 lyre00.cse.unsw.edu.au NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39778
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 10, ADDITIONAL: 15
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;lyre00.cse.unsw.edu.au.              IN      NS

;; AUTHORITY SECTION:
au.                172800  IN      NS      a.au.
au.                172800  IN      NS      b.au.
au.                172800  IN      NS      c.au.
au.                172800  IN      NS      d.au.
au.                172800  IN      NS      q.au.
au.                172800  IN      NS      r.au.
au.                172800  IN      NS      s.au.
au.                172800  IN      NS      t.au.
au.                172800  IN      NS      u.au.
au.                172800  IN      NS      v.au.

;; ADDITIONAL SECTION:
a.au.              172800  IN      A       58.65.254.73
b.au.              172800  IN      A       58.65.253.73
c.au.              172800  IN      A       162.159.24.179
d.au.              172800  IN      A       162.159.25.38
q.au.              172800  IN      A       65.22.196.1
r.au.              172800  IN      A       65.22.197.1
s.au.              172800  IN      A       65.22.198.1
t.au.              172800  IN      A       65.22.199.1
u.au.              172800  IN      A       211.29.133.32
v.au.              172800  IN      A       202.12.31.53
a.au.              172800  IN      AAAA    2407:6e00:254:306::73
b.au.              172800  IN      AAAA    2407:6e00:253:306::73
c.au.              172800  IN      AAAA    2400:cb00:2049:1::a29f:18b3
d.au.              172800  IN      AAAA    2400:cb00:2049:1::a29f:1926
q.au.              172800  IN      AAAA    2a01:8840:be::1

;; Query time: 208 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Wed Aug 15 13:58:03 2018
;; MSG SIZE  rcvd: 500
```

Thirdly, query 'edu.au.' domain:

```
z5195349@vx1:/tmp_amd/reed/export/reed/2/z5195349$ dig @58.65.254.73 lyre00.cse.unsw.edu.au NS

; <<>> DiG 9.7.3 <<>> @58.65.254.73 lyre00.cse.unsw.edu.au NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10728
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 8
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;lyre00.cse.unsw.edu.au.              IN      NS

;; AUTHORITY SECTION:
edu.au.            86400   IN      NS      s.au.
edu.au.            86400   IN      NS      q.au.
edu.au.            86400   IN      NS      r.au.
edu.au.            86400   IN      NS      t.au.

;; ADDITIONAL SECTION:
q.au.              86400   IN      A       65.22.196.1
r.au.              86400   IN      A       65.22.197.1
s.au.              86400   IN      A       65.22.198.1
t.au.              86400   IN      A       65.22.199.1
q.au.              86400   IN      AAAA    2a01:8840:be::1
r.au.              86400   IN      AAAA    2a01:8840:bf::1
s.au.              86400   IN      AAAA    2a01:8840:c0::1
t.au.              86400   IN      AAAA    2a01:8840:c1::1

;; Query time: 15 msec
;; SERVER: 58.65.254.73#53(58.65.254.73)
;; WHEN: Wed Aug 15 14:00:37 2018
;; MSG SIZE  rcvd: 280
```

And then, query 'unsw.edu.au.' domain:

```
z5195349@vx1:/tmp_amd/reed/export/reed/2/z5195349$ dig @65.22.196.1 lyre00.cse.unsw.edu.au NS

; <<>> DiG 9.7.3 <<>> @65.22.196.1 lyre00.cse.unsw.edu.au NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24548
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 5
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;lyre00.cse.unsw.edu.au.              IN      NS

;; AUTHORITY SECTION:
unsw.edu.au.            900      IN      NS      ns1.unsw.edu.au.
unsw.edu.au.            900      IN      NS      ns2.unsw.edu.au.
unsw.edu.au.            900      IN      NS      ns3.unsw.edu.au.

;; ADDITIONAL SECTION:
ns1.unsw.edu.au.        900      IN      A       129.94.0.192
ns2.unsw.edu.au.        900      IN      A       129.94.0.193
ns3.unsw.edu.au.        900      IN      A       192.155.82.178
ns1.unsw.edu.au.        900      IN      AAAA    2001:388:c:35::1
ns2.unsw.edu.au.        900      IN      AAAA    2001:388:c:35::2

;; Query time: 15 msec
;; SERVER: 65.22.196.1#53(65.22.196.1)
;; WHEN: Wed Aug 15 14:02:33 2018
;; MSG SIZE  rcvd: 198
```

Next, query 'cse.unsw.edu.au.' domain:

```
z5195349@vx1:/tmp_amd/reed/export/reed/2/z5195349$ dig @129.94.0.192 lyre00.cse.unsw.edu.au NS

; <<>> DiG 9.7.3 <<>> @129.94.0.192 lyre00.cse.unsw.edu.au NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44834
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 4
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;lyre00.cse.unsw.edu.au.              IN      NS

;; AUTHORITY SECTION:
cse.unsw.edu.au.        10800    IN      NS      beethoven.orchestra.cse.unsw.edu.au.
cse.unsw.edu.au.        10800    IN      NS      maestro.orchestra.cse.unsw.edu.au.

;; ADDITIONAL SECTION:
beethoven.orchestra.cse.unsw.edu.au. 10800 IN A 129.94.242.2
beethoven.orchestra.cse.unsw.edu.au. 10800 IN A 129.94.172.11
beethoven.orchestra.cse.unsw.edu.au. 10800 IN A 129.94.208.3
maestro.orchestra.cse.unsw.edu.au. 10800 IN A   129.94.242.33

;; Query time: 4 msec
;; SERVER: 129.94.0.192#53(129.94.0.192)
;; WHEN: Wed Aug 15 14:04:06 2018
;; MSG SIZE  rcvd: 160
```

Finally, we get 'cse.unsw.edu.au' domain's servers above. Therefore, we can query one of them to get IP address of host lyre00.cse.unsw.edu.au. (Note that we use 'A' to query the IP address, not use 'NS'.):

```
z5195349@vx1:/tmp_amd/reed/export/reed/2/z5195349$ dig @129.94.242.2 lyre00.cse.unsw.edu.au A

; <<>> DiG 9.7.3 <<>> @129.94.242.2 lyre00.cse.unsw.edu.au A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19396
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;lyre00.cse.unsw.edu.au.            IN      A

;; ANSWER SECTION:
lyre00.cse.unsw.edu.au. 3600     IN      A      129.94.210.20

;; AUTHORITY SECTION:
cse.unsw.edu.au.         3600     IN      NS      beethoven.orchestra.cse.unsw.edu.au.
cse.unsw.edu.au.         3600     IN      NS      maestro.orchestra.cse.unsw.edu.au.

;; ADDITIONAL SECTION:
maestro.orchestra.cse.unsw.edu.au. 3600 IN A    129.94.242.33
beethoven.orchestra.cse.unsw.edu.au. 3600 IN A  129.94.242.2

;; Query time: 0 msec
;; SERVER: 129.94.242.2#53(129.94.242.2)
;; WHEN: Wed Aug 15 14:06:55 2018
;; MSG SIZE  rcvd: 144
```

The IP address of lyre00.cse.unsw.edu.au is 129.94.210.20. We had to query 5 DNS servers (a.root-servers.net, a.au, q.au, ns1.unsw.edu.au, maestro.orchestra.cse.unsw.edu.au).

Question 11. Can one physical machine have several names and/or IP addresses associated with it?

Answer 11: Yes, one physical machine can have several names and several IP addresses. This is because a physical machine can have many interfaces, and one interfaces can have several IP addresses (May be including some virtual IP addresses or using NAT to make servers' different ports correspond to IP addresses (private addresses)). In addition, an IP address can have several hostnames.

(*) Exercise 4: A Simple Web Server

Note: Include your code in the lab report and demonstrate this exercise to your tutor in the lab slot.

In this exercise, you will learn the basics of TCP socket programming: how to create a socket, bind it to a specific address and port, as well as send and receive an HTTP packet. You will also learn some basics of HTTP header format. You will develop a web server that handles one HTTP request at a time. Specifically, your web server should do the following:

(i) create a connection socket when contacted by a client (browser).

(ii) receive HTTP request from this connection. Your server should only process GET request. You may assume that only GET requests will be received.

(iii) parse the request to determine the specific file being requested.

(iv) get the requested file from the server's file system.

(v) create an HTTP response message consisting of the requested file preceded by header lines.

(vi) send the response over the TCP connection to the requesting browser.

(vii) If the requested file is not present on the server, the server should send an HTTP "404 Not Found" message back to the client.

(viii) the server should listen in a loop, waiting for next request from the browser.

You don't have to deal with any other error conditions.

Your program should be called WebServer.c or WebServer.java or WebServer.py.

You should write the server so that it executes with the following command:

```
$java WebServer port (for Java)

$WebServer port (for C)

$python WebServer.py port (for Python)
```

where port is the port No your Web server will be listening on.

1. Place a simple HTML file (index.html, without any hyperlinks) in the same directory as the server program. A sample index.html file is provided here . Run the server program as indicated above. Open a web browser on the same machine. Type the following url: http://127.0.0.1:port/index.html where port is the port number the server listens on. If you forget to include the port number, the browser will assume the default port of 80. The browser should display the content of index.html.

2. Place multiple image files (.png) in the same directory as the server program. Run the server program as indicated above. Open a web browser on the same machine. Type http://127.0.0.1:port/myimage.png

where port is the port number the server listens on and myimage.png is one of the image files present in the server's directory. The browser should display the image.
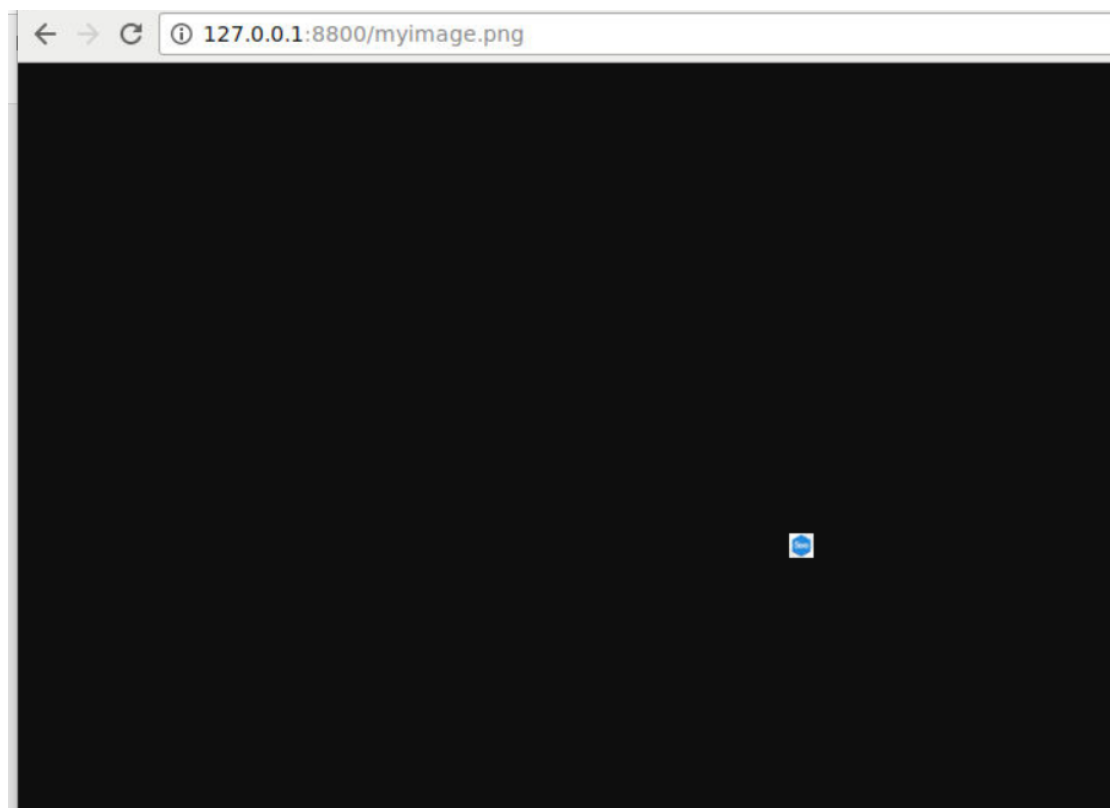
3. Now try and request for an object that does not exist in the server directory, e.g.: http://127.0.0.1:port/bio.html . The browser should display the 404 error message.

Note that you cannot use any of the pre-made web servers available in different programming languages.

Visit: http://*127.0.0.1:8800/index.html*

Visit: http://*127.0.0.1:8800/myimage.png*

Visit: http://*127.0.0.1:8800/bio.html*
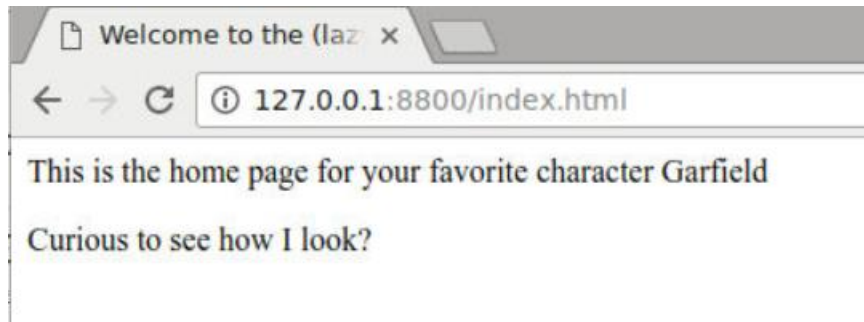
This is my code:

```python
import socket

import sys

from multiprocessing import Process




def handle_client(client_socket):

    """handle client connection request"""

    request_data = client_socket.recv(1024)

    print(request_data)

    request_lines = request_data.splitlines()

    for line in request_lines:

        print(line)

    # 'GET /filename HTTP/1.1'

    if len(request_lines[0]):

        request_start_line = request_lines[0].decode("utf-8")


    # get filename, GET /index.html HTTP/1.1

    get_file_name = str(request_start_line).split(' ')

    if get_file_name[0] != "GET":

        print("Incorrect request!")

        return

    else:

        file_name = get_file_name[1][1:]



# open file and read

        try:

            file = open(file_name, "rb")

        except IOError:
```

```python
            response_start_line = "HTTP/1.1 404 Not Found\r\n"

            response_heads = "Server: My python server\r\n"

            response_body = "The file not found!"

            response = response_start_line + response_heads + "\r\n" + response_body

            client_socket.sendall(bytes(response))

        else:

            file_data = file.read()

            file.close()


            response_start_line = "HTTP/1.1 200 OK\r\n"

            response_heads = "Server: My python server\r\n"

            if file_name.find('.html') > 0:                    # html or png

                print(file_name.find('.html'))

                response_content_type = "Content-Type:text/html\r\n"

                response_body = file_data.decode("utf-8")

                response = response_start_line + response_heads + response_content_type + "\r\n" +
response_body

                client_socket.sendall(bytes(response))

            else:

                response_content_type = "Content-Type:image/png\r\n"

                response_body = file_data

                response = response_start_line + response_heads + response_content_type + "\r\n"

                client_socket.sendall(bytes(response))

                client_socket.sendall(response_body)

        client_socket.close()



if __name__ == "__main__":

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)    # tcp socket

    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
```

```python
# host = socket.gethostname()   # get localhost

if len(sys.argv) == 2:

    port = int(sys.argv[1])

else:

    port = 80

s.bind(("", port))   # 127.0.0.1 port

s.listen(5)


while True:

    c, addr = s.accept()   # accept from client

    handle_client_process = Process(target=handle_client, args=(c,))   # build function

    handle_client_process.start()

    c.close()
```