# Iterative DNS Resolver

## Group Members:

| Name | Roll |
|---|---|
| Sakib Hasan | 149 |
| Md. Sakib Khan | 045 |

## Platform:

JAVA (Programming Language)

## Program Design:

In this assignment our task was to implement a simple DNS resolver that only issues *iterative queries* rather than using recursive queries. A *makefile* for the program was also asked to be included.

The solution code is written in Java programming language. The solution is contained in the file *IDNSresolver.java* which consists of one single class – IDNSresolver which contains several methods to perform the DNS query, get response and show in formatted output.

Firstly, *getDomain()* is the method where user input is taken. Use can give a hostname such as - "www.google.com".

After that there is a method *setDNSserver()* where all the root name servers are defined and set as the root server where the DNS query will be sent. This root name server is selected randomly from the 13 servers. In special case for hostname with the extension ".org" or ".edu" root server is set fixed as there are some issues while getting the response.

Then we have *sendDNSRequest(DatagramSocket, DNSAddress, domainName)* where we created the query message to be sent to root server. We chose a random socket number greater than 1023 as DNS uses UDP for its functionality.

To get response from the server this method- *getResponseFromServer(DatagramSocket)* where the response from the server is captured and formatted to be used to show in the terminal. *showDNSResponse(in)* takes the server response as DataInputStream and the response is formatted as readable format for the user. Type A and CNAME is handles in the program for the time being.

Furthermore, there are two more methods to convert ByteArray to IPv4 and IPv6. In the main function the output is formatted and shown as readable and understandable manner with proper details.

Note:
Moreover, there is also a makefile attached to run the program easily. To compile & run the program from terminal using makefile type command –> ***make dns***
To remove the .class file use command –> ***make clean***

Following are some example of testing the program (input & output):
For “www.google.com”

```
sakib2263@Classified-006:/mnt/c/IDNSresolver$ make dns
javac IDNSresolver.java
java  IDNSresolver
Enter the Hostame (web address) :
www.google.com
RootServer = d.root-servers.net
Given Domain name: www.google.com
Root NS address: 199.7.91.13 #53
-------------------------------------------------------------------------------------------
Sending Query to IP: 199.7.91.13
Sent 32 bytes (Query Size)


Received Data length: 48 bytes
-------------------------------------------------------------------------------------------
* Transaction ID: (1234)
* Flags: 0x8080
* Questions: 1
* Answers RR(s): 1
* Authority RR(s): 0
* Additional RR(s): 0
Query Domain: www.google.com
------------------------------------ ANSWER RR(s) -------------------------------------

Name: 0xc00c
Type: A
Class: IN
TTL: 210
Length: 4 byte(s)
IP Address: 172.217.163.132


-------------------------------------------------------------------------------------------
For domain :www.google.com ---> IP Address: 172.217.163.132
```

Here, Hostname is taken from user input, then root server is randomly selected and the query is sent. The response from the server is shown in the Standard DNS response format.
All the answer RR(s) information are shown in the “ANSWER RR(s)” section along with all the IP addresses for the hostname. Finally one of the IP for the domain is shown.

Let’s check some other DNS query and responses out –
For “microsoft.com”:

```
Enter the Hostame (web address) :
microsoft.com
Selected RootServer = d.root-servers.net
Given Domain name: microsoft.com
Root NS address: 199.7.91.13 #53
-------------------------------------------------------------------------------------------
Sending Query to IP: 199.7.91.13
Sent 31 bytes (Query Size)


Received Data length: 111 bytes
-------------------------------------------------------------------------------------------
* Transaction ID: (1234)
* Flags: 0x8080
* Questions: 1
* Answers RR(s): 5
* Authority RR(s): 0
* Additional RR(s): 0
Query Domain: microsoft.com
------------------------------------- ANSWER RR(s) -------------------------------------

Name: 0xc00c
Type: A
Class: IN
TTL: 3522
Length: 4 byte(s)
IP Address: 40.112.72.205


Name: 0xc00c
Type: A
Class: IN
TTL: 3522
Length: 4 byte(s)
IP Address: 40.113.200.201


Name: 0xc00c
Type: A
Class: IN
TTL: 3522
Length: 4 byte(s)
IP Address: 104.215.148.63


Name: 0xc00c
Type: A
Class: IN
TTL: 3522
Length: 4 byte(s)
IP Address: 13.77.161.179


Name: 0xc00c
Type: A
Class: IN
TTL: 3522
Length: 4 byte(s)
IP Address: 40.76.4.15


-------------------------------------------------------------------------------------------
For domain :microsoft.com ---> IP Address: 40.112.72.205
```

For ‘microsoft.com’ we had 4 answers in the response section and corresponding 4 different IP addresses as well.

Now, check for the domain www.nobody.com which actually do not exist.

```
sakib2263@Classified-006:/mnt/c/IDNSresolver$ make dns
javac IDNSresolver.java
java  IDNSresolver
Enter the Hostame (web address) :
www.nobody.com
Selected RootServer = j.root-servers.net
Given Domain name: www.nobody.com
Root NS address: 192.58.128.30 #53
---------------------------------------------------------------------------------------------
Sending Query to IP: 192.58.128.30
Sent 32 bytes (Query Size)


Received Data length: 32 bytes
---------------------------------------------------------------------------------------------
* Transaction ID: (1234)
* Flags: 0x8082
* Questions: 1
* Answers RR(s): 0
* Authority RR(s): 0
* Additional RR(s): 0
Query Domain: www.nobody.com
------------------------------------ ANSWER RR(s) -------------------------------------
Server DNS address could not be found!



---------------------------------------------------------------------------------------------

www.nobody.com Does not exist!
```

So there is no answer in the response from server and no IP address found. Thus DNS for this domain was not resolved and that’s why “www.nobody.com does not exist!” message is shown.

**Limitations:**

Thread was not used in this program and we did not wait for a certain amount to get response from the server. So, for some cases while retrieving domain type with extension “.edu” this program cannot resolve DNS.