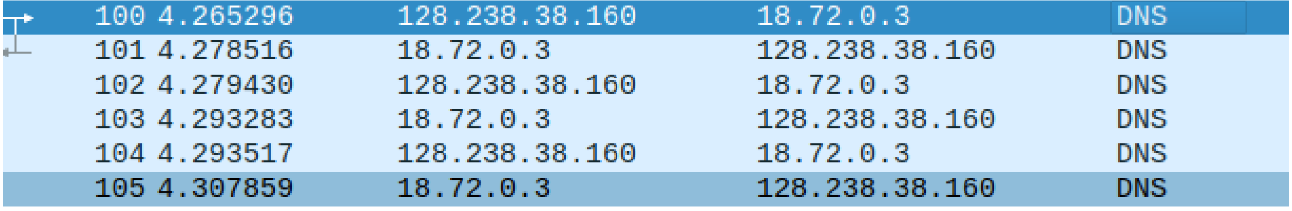
DCCN LAB TEST

SET 1

Name: MALOTH ADITYA
Roll No.: 120CS0124

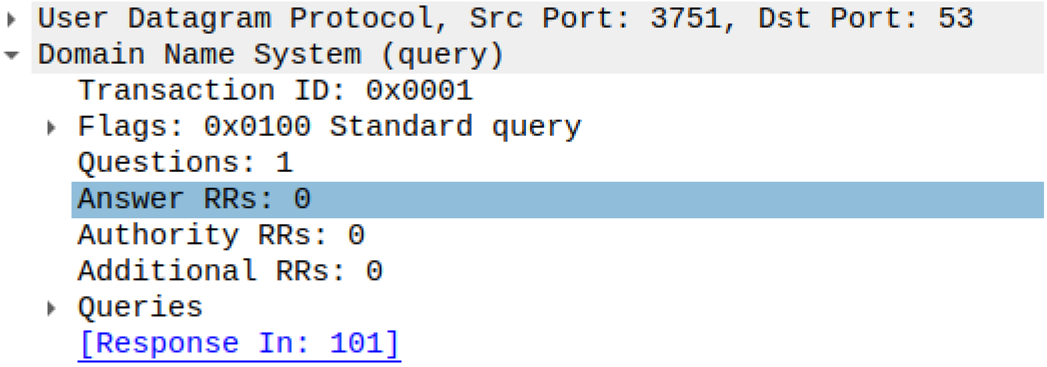
1.

a. DNS query is sent to **18.72.0.3.**
The ip address cannot be the local DNS server.



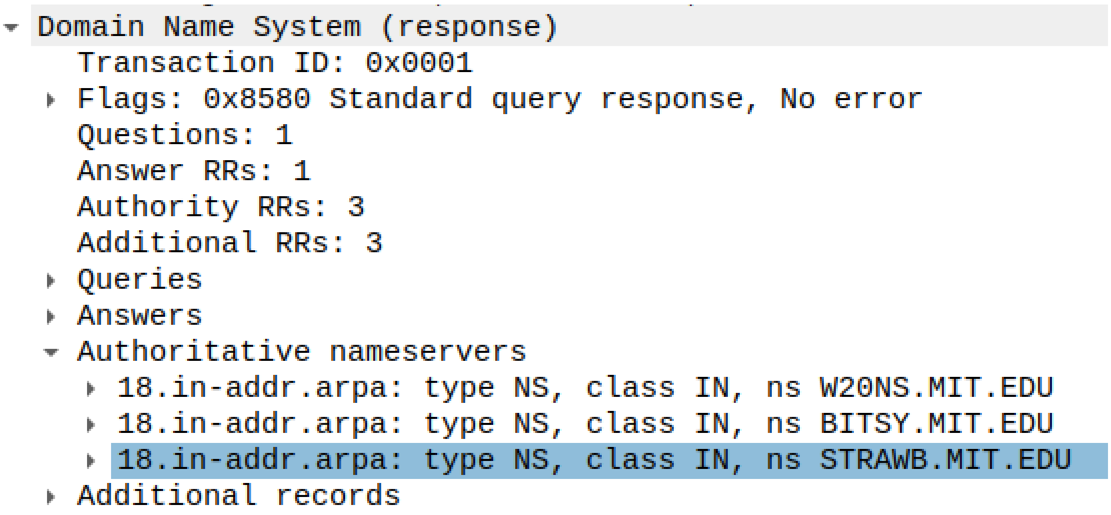
```
100 4.265296   128.238.38.160   18.72.0.3        DNS
101 4.278516   18.72.0.3        128.238.38.160   DNS
102 4.279430   128.238.38.160   18.72.0.3        DNS
103 4.293283   18.72.0.3        128.238.38.160   DNS
104 4.293517   128.238.38.160   18.72.0.3        DNS
105 4.307859   18.72.0.3        128.238.38.160   DNS
```

b. The DNS message type is Standard query.
The query message does not contain any answers



```
▸ User Datagram Protocol, Src Port: 3751, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x0001
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▸ Queries
    [Response In: 101]
```

c. The DNS response provides following nameservers:
W20NS.MIT.EDU
BITSY.MIT.EDU
STRAWB.MIT.EDU

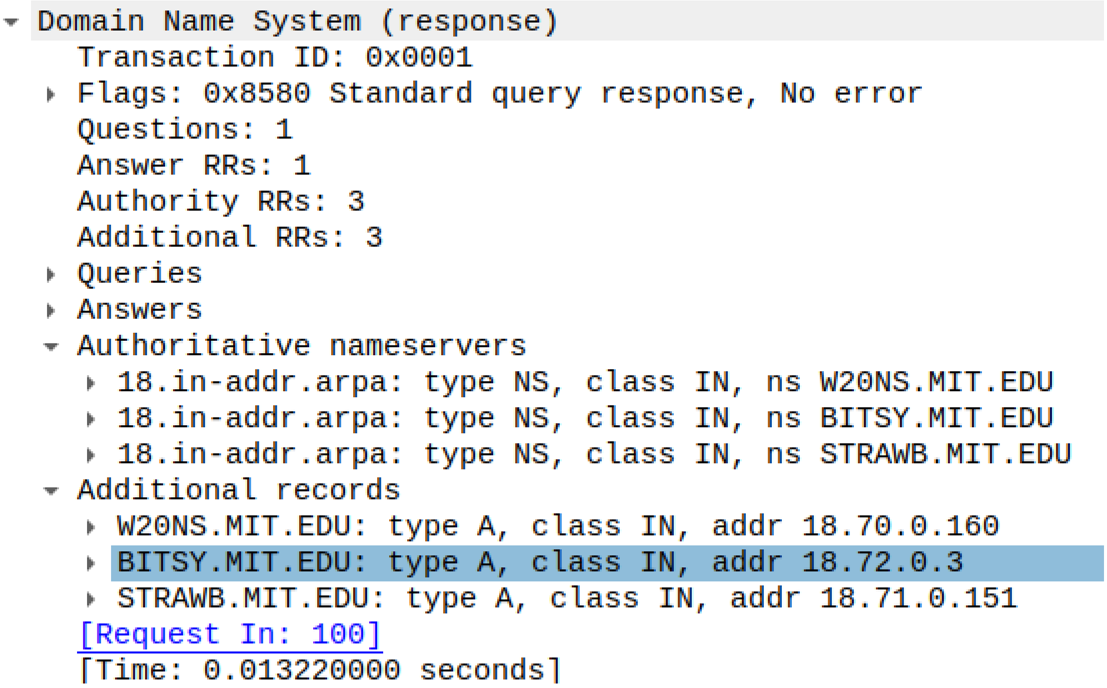


```
▾ Domain Name System (response)
    Transaction ID: 0x0001
  ▸ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 3
  ▸ Queries
  ▸ Answers
  ▾ Authoritative nameservers
    ▸ 18.in-addr.arpa: type NS, class IN, ns W20NS.MIT.EDU
    ▸ 18.in-addr.arpa: type NS, class IN, ns BITSY.MIT.EDU
    ▸ 18.in-addr.arpa: type NS, class IN, ns STRAWB.MIT.EDU
  ▸ Additional records
```

DCCN LAB TEST

d. Yes, the response message provides ip addresses of the nameservers

W20NS.MIT.EDU	18.70.0.160
BITSY.MIT.EDU	18.72.0.3
STRAWB.MIT.EDU	18.71.0.151



```
▾ Domain Name System (response)
    Transaction ID: 0x0001
  ▸ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 3
  ▸ Queries
  ▸ Answers
  ▾ Authoritative nameservers
    ▸ 18.in-addr.arpa: type NS, class IN, ns W20NS.MIT.EDU
    ▸ 18.in-addr.arpa: type NS, class IN, ns BITSY.MIT.EDU
    ▸ 18.in-addr.arpa: type NS, class IN, ns STRAWB.MIT.EDU
  ▾ Additional records
    ▸ W20NS.MIT.EDU: type A, class IN, addr 18.70.0.160
    ▸ BITSY.MIT.EDU: type A, class IN, addr 18.72.0.3
    ▸ STRAWB.MIT.EDU: type A, class IN, addr 18.71.0.151
    [Request In: 100]
    [Time: 0.013220000 seconds]
```

2.

TCP SERVER:

```
//TCP Server program
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main(){
	int serid,sessid,num1,num2;
	struct sockaddr_in server_address,client_address;

	unsigned int server_addlen,client_addlen;

	server_address.sin_family=AF_INET;
	server_address.sin_addr.s_addr=inet_addr("127.0.0.1");
	server_address.sin_port=9876;

	server_addlen=sizeof(server_address);
	client_addlen=sizeof(client_addlen);
	serid=socket(AF_INET,SOCK_STREAM,0);

	bind(serid,(struct sockaddr*)&server_address,server_addlen);

	listen(serid,10);

	while(1){
		printf("Server is ready to accept ......\n");
		sessid=accept(serid,(struct sockaddr
*)&client_address,&client_addlen);
		read(sessid,&num1,4);
		read(sessid,&num2,4);
		printf("\nMessage received from client\nThe numbers are: %d %d\n\
n",num1,num2);
		num1 = num1/num2;
		write(sessid,&num1,4);
		close(sessid);
	}
	return(0);
}
```

TCP CLIENT:

```
//TCP Client program
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main(){
	int sid,num1,num2;
	struct sockaddr_in server_address;
	int server_addlen;

	server_address.sin_family=AF_INET;
	char ipAddr[15];
	printf("Enter the server IP address: ");
	gets(ipAddr);

	server_address.sin_addr.s_addr=inet_addr(ipAddr);

	server_address.sin_port=9876;

	server_addlen=sizeof(server_address);
```

DCCN LAB TEST

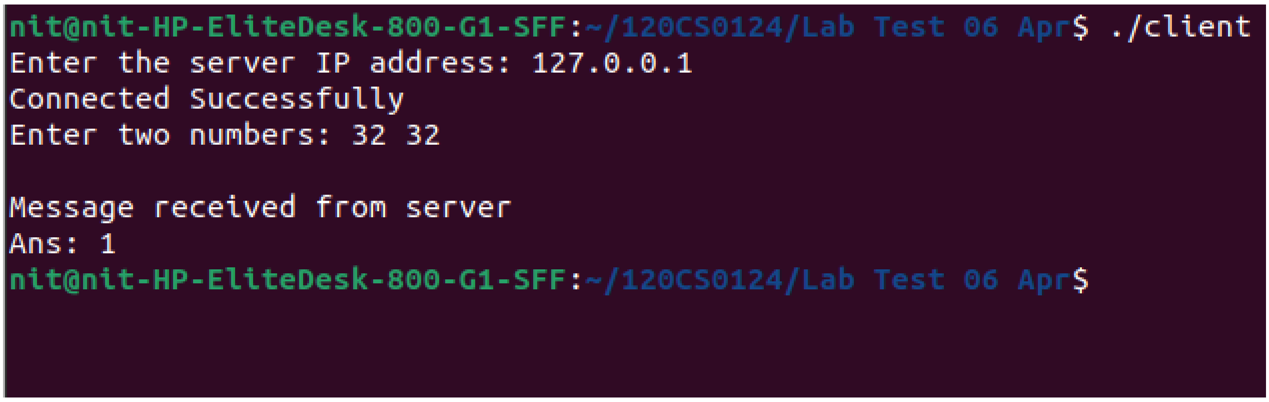
```
        sid=socket(AF_INET,SOCK_STREAM,0);

        connect(sid,(struct sockaddr *)&server_address,server_addlen);
        printf("Connected Successfully\nEnter two numbers: ");
        scanf("%d %d",&num1,&num2);
        write(sid,&num1,4);
        write(sid,&num2,4);
        read(sid,&num1,1);

        printf("\nMessage received from server\nAns: %d\n",num1);
        close(sid);
        return(0);
}
```

OUTPUT:

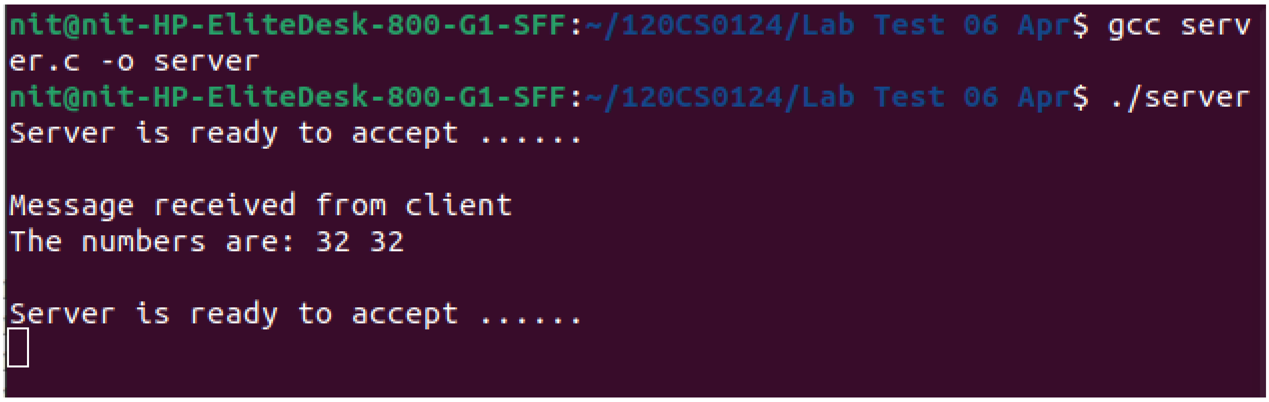
CLIENT:



```
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab Test 06 Apr$ ./client
Enter the server IP address: 127.0.0.1
Connected Successfully
Enter two numbers: 32 32

Message received from server
Ans: 1
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab Test 06 Apr$
```

SERVER:



```
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab Test 06 Apr$ gcc serv
er.c -o server
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab Test 06 Apr$ ./server
Server is ready to accept ......

Message received from client
The numbers are: 32 32

Server is ready to accept ......
```