EX7 – DOMAIN NAME SERVER USING UDP

- S. Vishakan CSE – C 18 5001 196

Server Program:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

#define PORT 7228
#define SIZE 30

struct DNS_Table{ //struct for the DNS table
        char server_list[SIZE][100];
        char ip_list[SIZE][50];
        int cur_size;
};

typedef struct DNS_Table dns_table;

dns_table *local, *root, *auth; //local - 1st level, root - 2nd level, auth - 3rd level

char empty[5] = "NULL\0";

int     checkIP(char *ip);
int     addRecord(dns_table *table, char *server, char *ip);
void    allocateIP(dns_table *table);
void    printTable(dns_table *table, char *table_name);
void    updateTable(dns_table *table);
void    initTables();
char    *fetchIP(char *req_server);

int main(int argc, char **argv){

        struct sockaddr_in server, client;
        int sockfd, n, addrlen, flag;
        char req_server[100], *ip, req_ip[50];

        initTables();
        printTable(local, "Local Table");
        printTable(root, "Root Table");
        printTable(auth, "Authoritative Table");
```

```
printf("\n\nDo you wish to alter the local allocation table? (1 - YES, 0 - NO) -> ");
scanf("%d", &flag);

if(flag == 1){
        updateTable(local);
        printTable(local, "Local Table");
}

sockfd = socket(AF_INET, SOCK_DGRAM, 0);

if(sockfd < 0){
        perror("Error in creating socket.\n");
}

bzero(&server, sizeof(server));
server.sin_family = AF_INET;
server.sin_addr.s_addr = INADDR_ANY;
server.sin_port = htons(PORT);

if(bind(sockfd, (struct sockaddr*)&server, sizeof(server)) < 0){
        perror("Error in binding.\n");
}

printf("\nServer awaiting clients on port %d...\n", PORT);

addrlen = sizeof(client);

while(1){
        bzero(req_server, sizeof(req_server));
        recvfrom(sockfd, &req_server, sizeof(req_server), 0, (struct
        sockaddr*)&client, &addrlen);
        printf("\nReceived a request for IP Address of %s from a client.\n",
        req_server);


        ip = fetchIP(req_server);

        if(ip == NULL){ //IP address does not exist
                strcpy(req_ip, empty);
                sendto(sockfd, &empty, sizeof(empty), 0, (struct sockaddr*)&client,
                addrlen);
        }
        else{
                //pointer -> char_array conversion is necessary since pointer only
                sends 8 bits of data
                strcpy(req_ip, ip);
                sendto(sockfd, &req_ip, sizeof(req_ip), 0, (struct sockaddr*)&client,
                addrlen);
        }
```

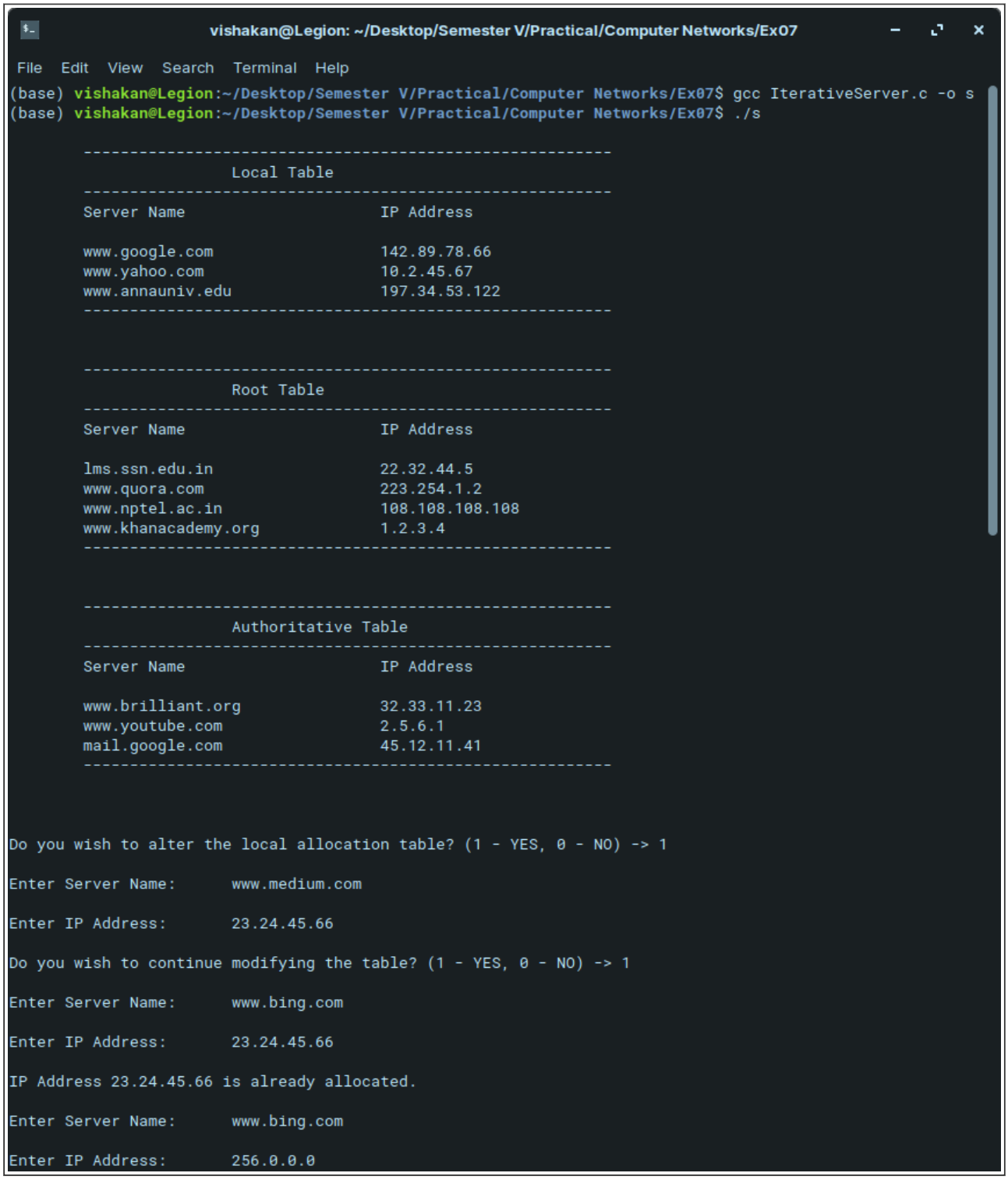
```
                printf("\n\nReplied with IP Address %s\n", req_ip);
        }

        return 0;
}

int checkIP(char *ip){
        //Checks for the validity of a given IP address

        int valid = 1, byte;
        char *ip_copy, *split;

        ip_copy = (char *)calloc(50, sizeof(char));
        strcpy(ip_copy, ip);
        split = strtok(ip_copy, ".");

        while(split){ //split pointer points to each "byte" iteratively
                byte = atoi(split);
                if(byte < 0 || byte > 255){
                        return 0;
                }

                split = strtok(NULL, ".");
        }

        return 1;
}

int addRecord(dns_table *table, char *server, char *ip){
        //Add a new record into a specific DNS table

        int valid;

        if(table->cur_size == SIZE - 1){ //if table is full
                return table->cur_size;
        }

        valid = checkIP(ip);

        if(valid){
                strcpy(table->server_list[table->cur_size], server);
                strcpy(table->ip_list[table->cur_size], ip);
                table->cur_size++;
        }

        else{
                printf("\tIP Address %s is invalid.\n", ip);
        }

        return table->cur_size;
}
```

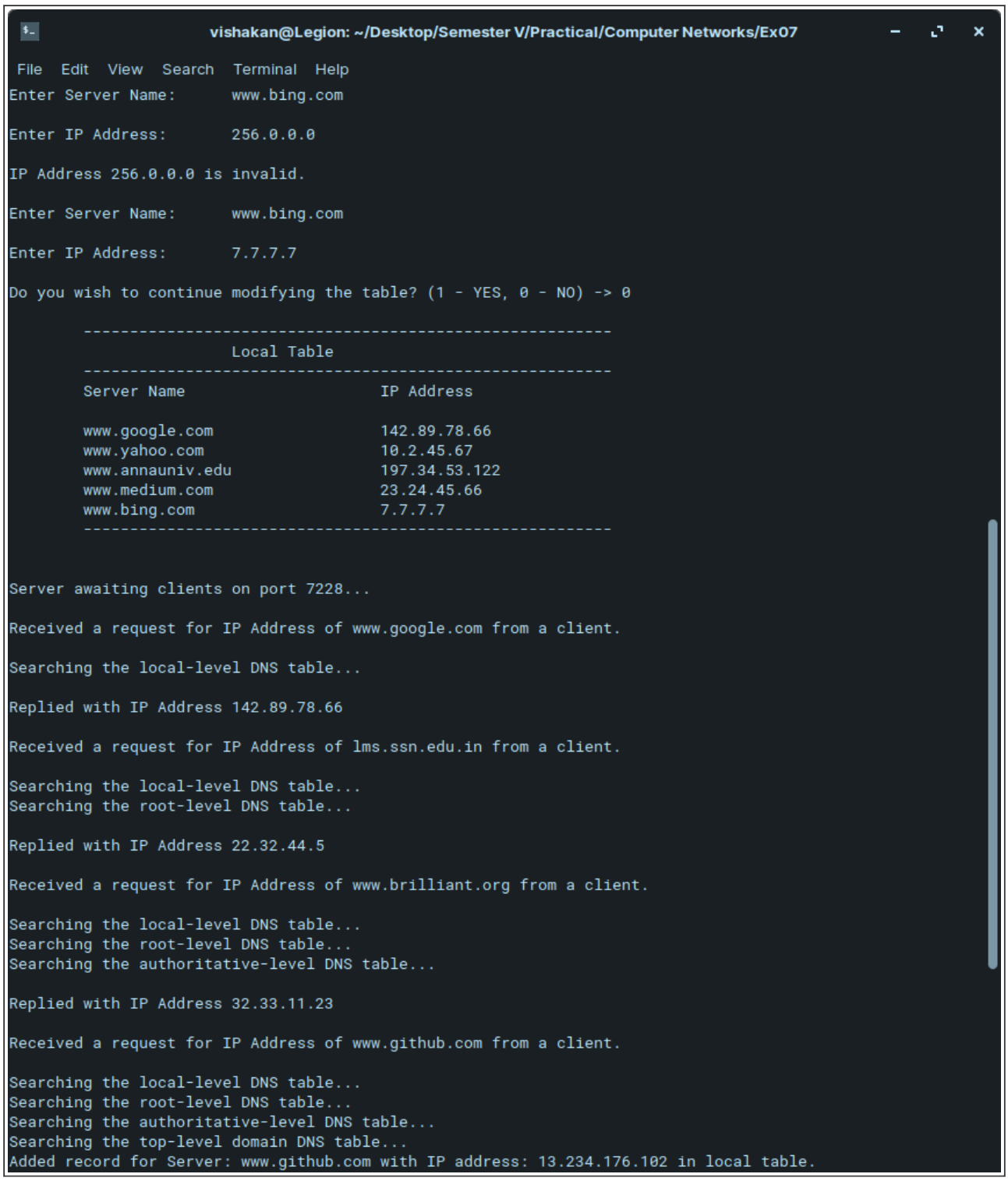
```
void printTable(dns_table *table, char *table_name){
	//Print the current contents of a given table

	int i = 0;

	printf("\n\t-------------------------------------------------------");
	printf("\n\t\t\t%-30s", table_name);
	printf("\n\t-------------------------------------------------------");
	printf("\n\t%-25s\t%s\n", "Server Name", "IP Address");

	for(i = 0; i < table->cur_size; i++){
		printf("\n\t%-25s\t%s", table->server_list[i], table->ip_list[i]);
	}
	printf("\n\t-------------------------------------------------------\n\n");
}

void updateTable(dns_table *table){
	//Update a given DNS table

	char serv[100], ip[50];
	int i = 0, exists = 0, choice = 1, valid;

	while(choice){
		printf("\nEnter Server Name:\t");
		scanf("%s", serv);
		printf("\nEnter IP Address:\t");
		scanf("%s", ip);

		valid = checkIP(ip);

		if(!valid){
			printf("\nIP Address %s is invalid.\n", ip);
			continue;
		}

		exists = 0;

		for(i = 0; i < table->cur_size; i++){
			if(strcmp(ip, table->ip_list[i]) == 0){
				exists = 1;
				printf("\nIP Address %s is already allocated.\n", ip);
				break;
			}
		}

		if(exists == 0){
			strcpy(table->ip_list[i], ip);
			strcpy(table->server_list[i], serv);
			table->cur_size++;
```

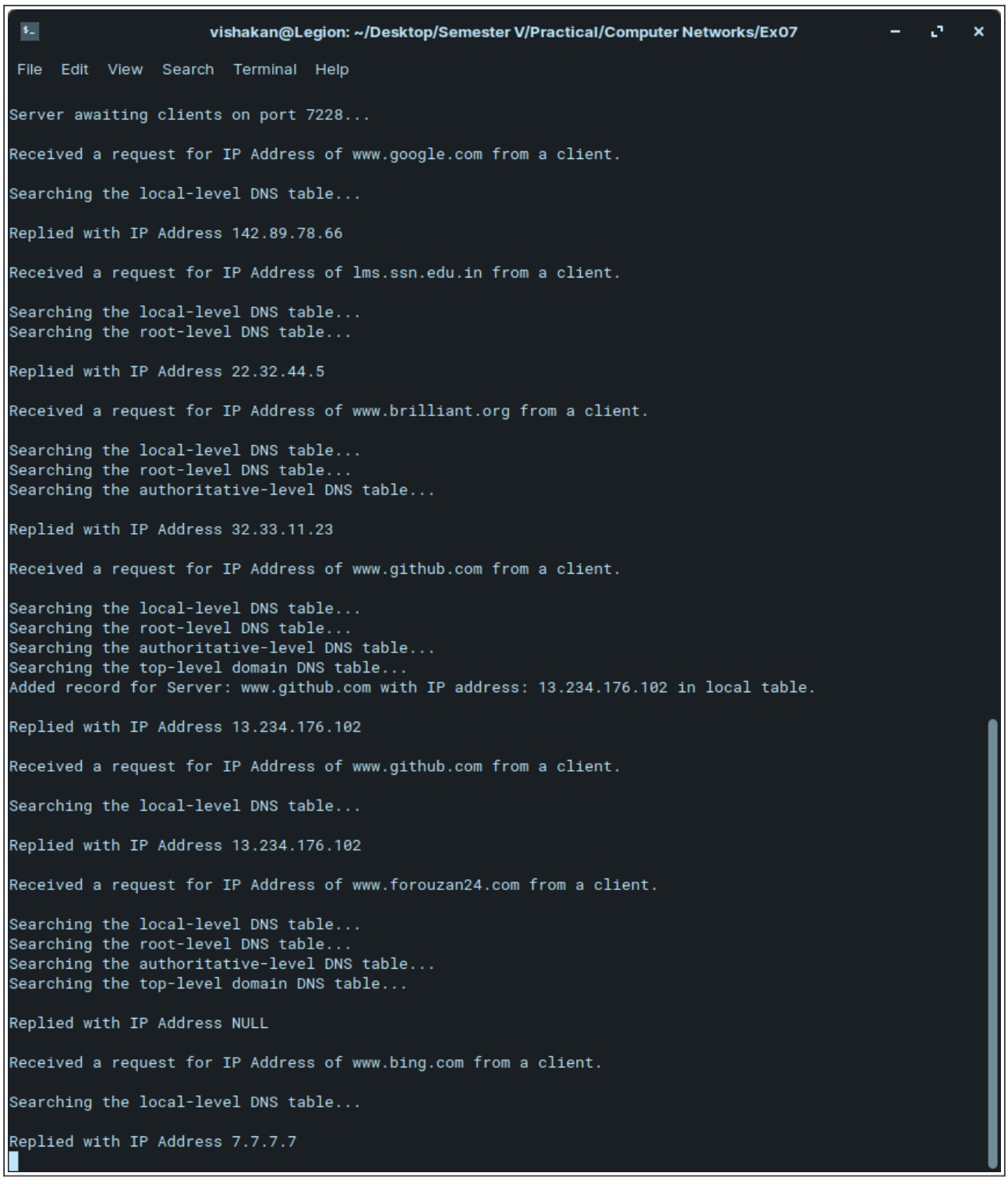
```
                printf("\nDo you wish to continue modifying the table? (1 - YES, 0 - NO)
                -> ");
                scanf("%d", &choice);
            }
        }
}

void initTables(){
        //Initialize the local, root and auth tables with some prefixed records

        local = (dns_table *)malloc(sizeof(dns_table));
        root = (dns_table *)malloc(sizeof(dns_table));
        auth = (dns_table *)malloc(sizeof(dns_table));

        local->cur_size = 0;
        root->cur_size = 0;
        auth->cur_size = 0;

        addRecord(local, "www.google.com", "142.89.78.66");
        addRecord(local, "www.yahoo.com", "10.2.45.67");
        addRecord(local, "www.annauniv.edu", "197.34.53.122");

        addRecord(root, "lms.ssn.edu.in", "22.32.44.5");
        addRecord(root, "www.quora.com", "223.254.1.2");
        addRecord(root, "www.nptel.ac.in", "108.108.108.108");
        addRecord(root, "www.khanacademy.org", "1.2.3.4");

        addRecord(auth, "www.brilliant.org", "32.33.11.23");
        addRecord(auth, "www.youtube.com", "2.5.6.1");
        addRecord(auth, "mail.google.com", "45.12.11.41");

}

char *fetchIP(char *req_server){
        //Fetch the IP address of a given domain name from the DNS tables iteratively/from
        TLD

        int i = 0;
        struct hostent *he; //for TLD
        char *tld_ip;

        printf("\nSearching the local-level DNS table...");

        for(i = 0; i < local->cur_size; i++){
            if(strcmp(local->server_list[i], req_server) == 0){
                return local->ip_list[i]; //found in local
            }
        }
```

```
        printf("\nSearching the root-level DNS table...");

        for(i = 0; i < root->cur_size; i++){
                if(strcmp(root->server_list[i], req_server) == 0){
                        return root->ip_list[i]; //found in root
                }
        }

        printf("\nSearching the authoritative-level DNS table...");

        for(i = 0; i < auth->cur_size; i++){
                if(strcmp(auth->server_list[i], req_server) == 0){
                        return auth->ip_list[i]; //found in auth
                }
        }

        printf("\nSearching the top-level domain DNS table...");

        he = gethostbyname(req_server); //gets the host entry for the domain name

        if(he == NULL){ //if host entry does not exist for the domain
                return NULL;
        }

        tld_ip = inet_ntoa(*((struct in_addr *)he->h_addr_list[0])); //get the IP address from
                                                                     host entry

        addRecord(local, req_server, tld_ip); //cache the record in local for faster access
                                              next time

        printf("\nAdded record for Server: %s with IP address: %s in local table.", req_server,
        tld_ip);


        return tld_ip;

}
```

Output:


vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex07
File Edit View Search Terminal Help
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$ gcc IterativeServer.c -o s
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$ ./s
Local Table
Server Name IP Address
www.google.com 142.89.78.66
www.yahoo.com 10.2.45.67
www.annauniv.edu 197.34.53.122
Root Table
Server Name IP Address
lms.ssn.edu.in 22.32.44.5
www.quora.com 223.254.1.2
www.nptel.ac.in 108.108.108.108
www.khanacademy.org 1.2.3.4
Authoritative Table
Server Name IP Address
www.brilliant.org 32.33.11.23
www.youtube.com 2.5.6.1
mail.google.com 45.12.11.41
Do you wish to alter the local allocation table? (1 - YES, 0 - NO) -> 1
Enter Server Name: www.medium.com
Enter IP Address: 23.24.45.66
Do you wish to continue modifying the table? (1 - YES, 0 - NO) -> 1
Enter Server Name: www.bing.com
Enter IP Address: 23.24.45.66
IP Address 23.24.45.66 is already allocated.
Enter Server Name: www.bing.com
Enter IP Address: 256.0.0.0

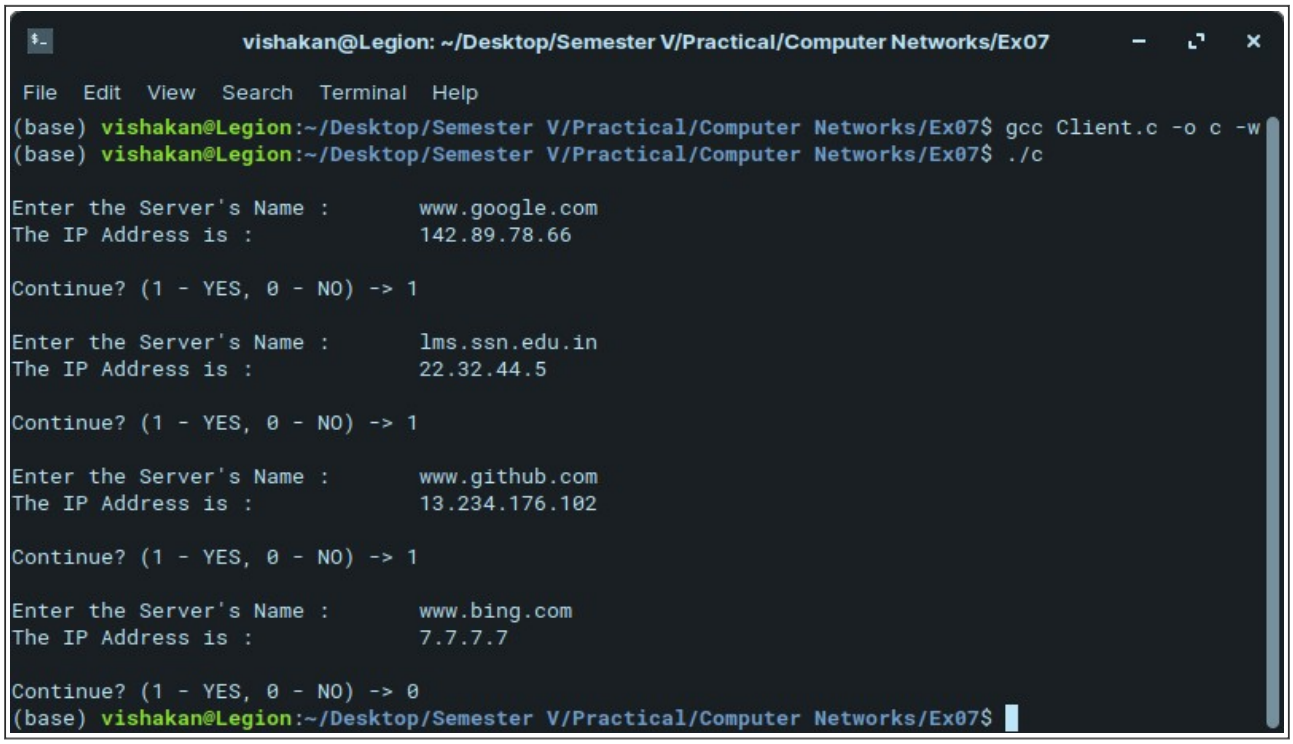

vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex07
File Edit View Search Terminal Help
Enter Server Name: www.bing.com
Enter IP Address: 256.0.0.0
IP Address 256.0.0.0 is invalid.
Enter Server Name: www.bing.com
Enter IP Address: 7.7.7.7
Do you wish to continue modifying the table? (1 - YES, 0 - NO) -> 0
Local Table
Server Name IP Address
www.google.com 142.89.78.66
www.yahoo.com 10.2.45.67
www.annauniv.edu 197.34.53.122
www.medium.com 23.24.45.66
www.bing.com 7.7.7.7
Server awaiting clients on port 7228...
Received a request for IP Address of www.google.com from a client.
Searching the local-level DNS table...
Replied with IP Address 142.89.78.66
Received a request for IP Address of lms.ssn.edu.in from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Replied with IP Address 22.32.44.5
Received a request for IP Address of www.brilliant.org from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Searching the authoritative-level DNS table...
Replied with IP Address 32.33.11.23
Received a request for IP Address of www.github.com from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Searching the authoritative-level DNS table...
Searching the top-level domain DNS table...
Added record for Server: www.github.com with IP address: 13.234.176.102 in local table.

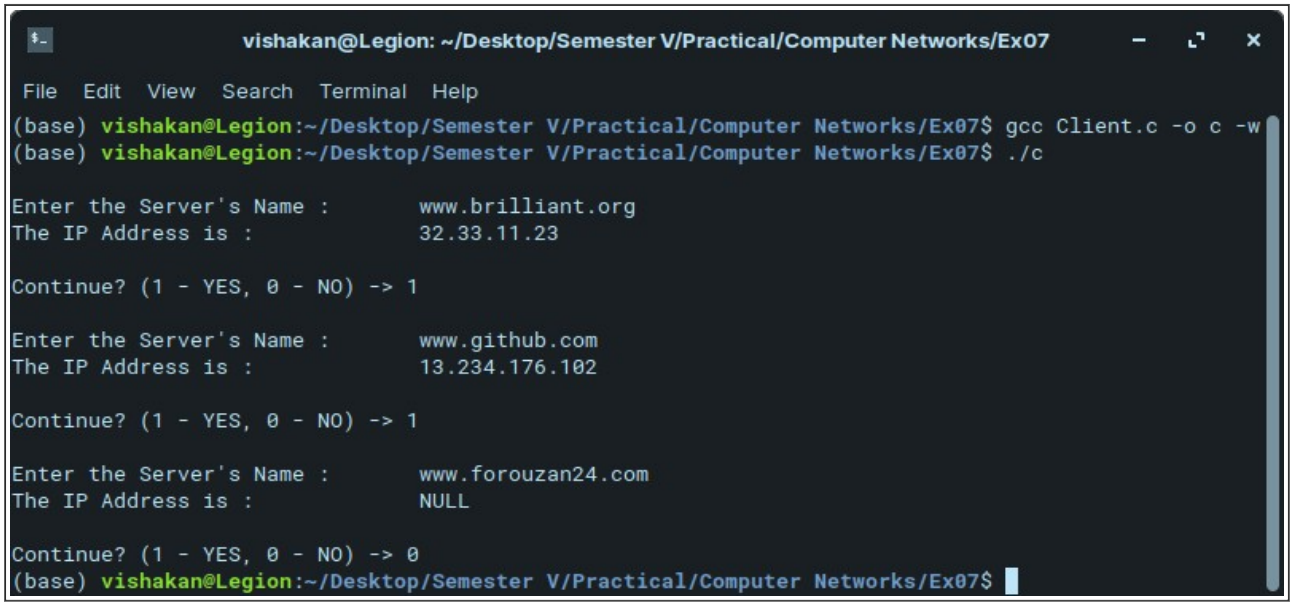

vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex07
File Edit View Search Terminal Help
Server awaiting clients on port 7228...
Received a request for IP Address of www.google.com from a client.
Searching the local-level DNS table...
Replied with IP Address 142.89.78.66
Received a request for IP Address of lms.ssn.edu.in from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Replied with IP Address 22.32.44.5
Received a request for IP Address of www.brilliant.org from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Searching the authoritative-level DNS table...
Replied with IP Address 32.33.11.23
Received a request for IP Address of www.github.com from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Searching the authoritative-level DNS table...
Searching the top-level domain DNS table...
Added record for Server: www.github.com with IP address: 13.234.176.102 in local table.
Replied with IP Address 13.234.176.102
Received a request for IP Address of www.github.com from a client.
Searching the local-level DNS table...
Replied with IP Address 13.234.176.102
Received a request for IP Address of www.forouzan24.com from a client.
Searching the local-level DNS table...
Searching the root-level DNS table...
Searching the authoritative-level DNS table...
Searching the top-level domain DNS table...
Replied with IP Address NULL
Received a request for IP Address of www.bing.com from a client.
Searching the local-level DNS table...
Replied with IP Address 7.7.7.7

Client Program:

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define PORT 7228

int main(int argc, char **argv){

	struct sockaddr_in server, client;
	int sockfd, n, addrlen, flag, choice = 1;
	char req_server[100], req_ip[50];

	sockfd = socket(AF_INET, SOCK_DGRAM, 0);

	if(sockfd < 0){
		perror("Error in creating socket.\n");
	}

	bzero(&server, sizeof(server));
	server.sin_family = AF_INET;
	server.sin_addr.s_addr = INADDR_ANY;
	server.sin_port = htons(PORT);

	addrlen = sizeof(server);

	while(choice){
		printf("\nEnter the Server's Name :\t");
		scanf("%s", req_server);
		sendto(sockfd, &req_server, sizeof(req_server), 0, (struct sockaddr*)&server,
		sizeof(server));

		recvfrom(sockfd, &req_ip, sizeof(req_ip), 0, (struct sockaddr*)&server,
		&addrlen);
		printf("The IP Address is :\t\t%s\n", req_ip);

		printf("\nContinue? (1 - YES, 0 - NO) -> ");
		scanf("%d", &choice);
	}

	close(sockfd);

	return 0;
}
```

Output:


vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex07
File Edit View Search Terminal Help
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$ gcc Client.c -o c -w
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$ ./c
Enter the Server's Name : www.google.com
The IP Address is : 142.89.78.66
Continue? (1 - YES, 0 - NO) -> 1
Enter the Server's Name : lms.ssn.edu.in
The IP Address is : 22.32.44.5
Continue? (1 - YES, 0 - NO) -> 1
Enter the Server's Name : www.github.com
The IP Address is : 13.234.176.102
Continue? (1 - YES, 0 - NO) -> 1
Enter the Server's Name : www.bing.com
The IP Address is : 7.7.7.7
Continue? (1 - YES, 0 - NO) -> 0
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$


vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex07
File Edit View Search Terminal Help
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$ gcc Client.c -o c -w
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$ ./c
Enter the Server's Name : www.brilliant.org
The IP Address is : 32.33.11.23
Continue? (1 - YES, 0 - NO) -> 1
Enter the Server's Name : www.github.com
The IP Address is : 13.234.176.102
Continue? (1 - YES, 0 - NO) -> 1
Enter the Server's Name : www.forouzan24.com
The IP Address is : NULL
Continue? (1 - YES, 0 - NO) -> 0
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex07$