1. **PROGRAM :**

#show ip configuration

Ipconfig

#view active TCP/UDP connection and port

Netstat -an

#perform dns lookup

Nslookup www.google.com

#trace a route to website

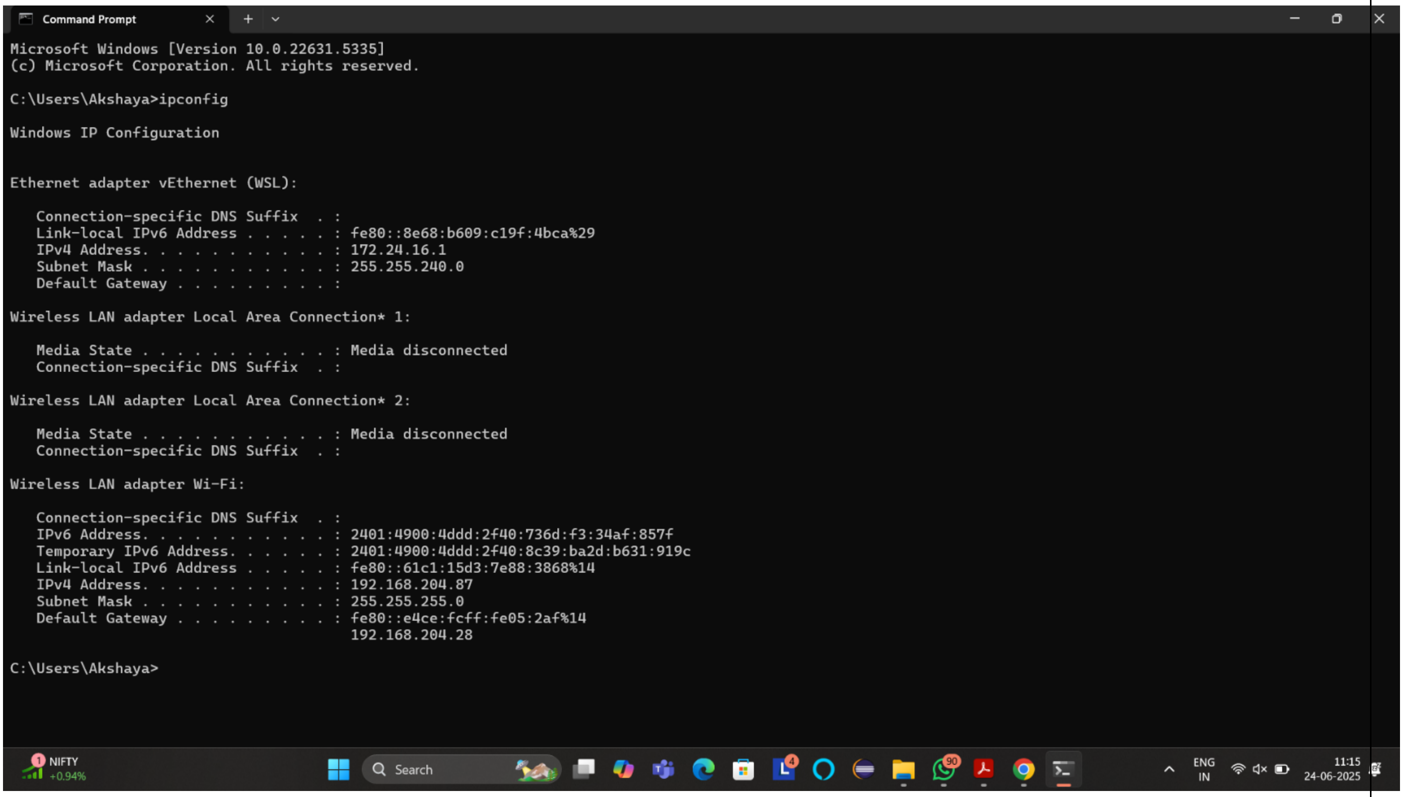
Tracert www.google.com

#ping server to test connectivity

Ping www.google.com

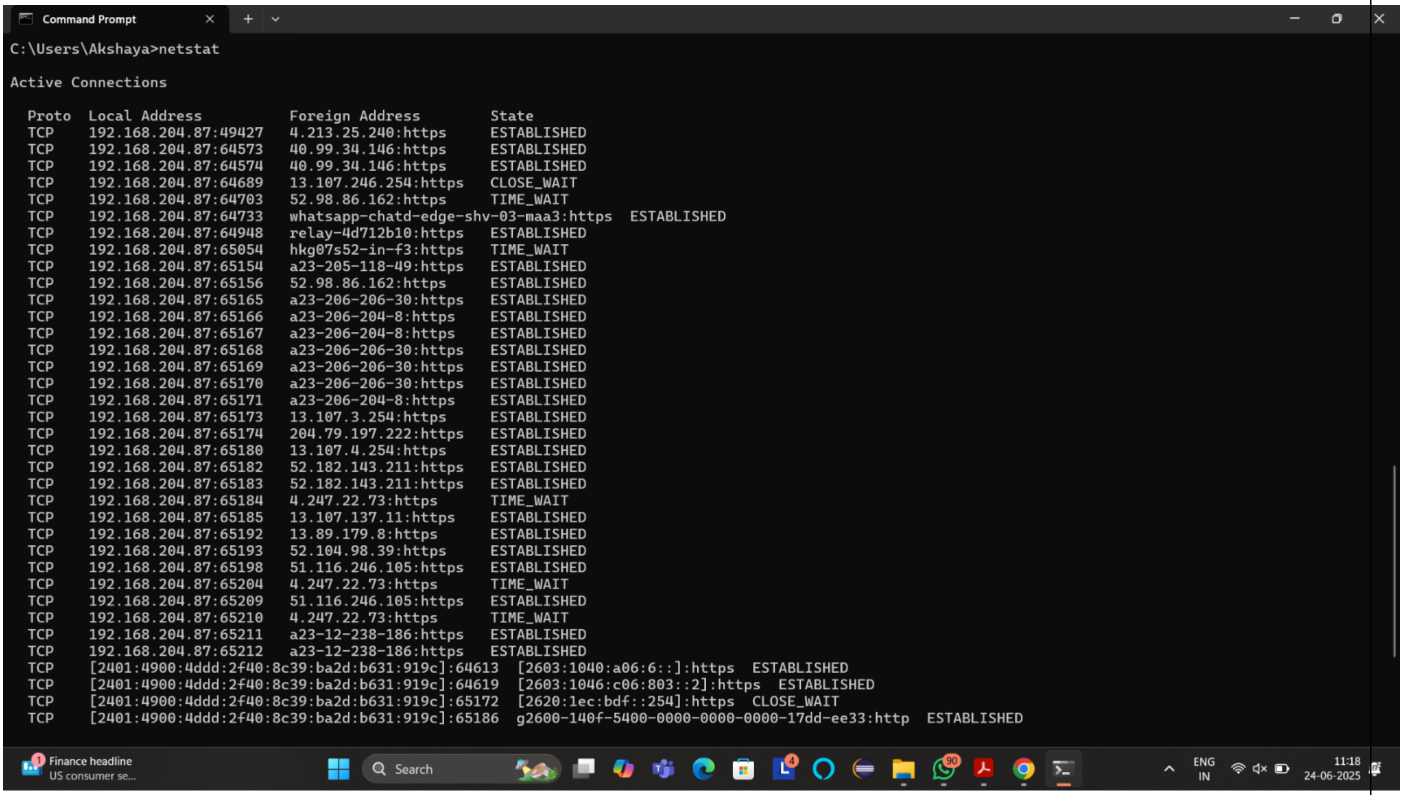
OUTPUT :

1.ipconfig



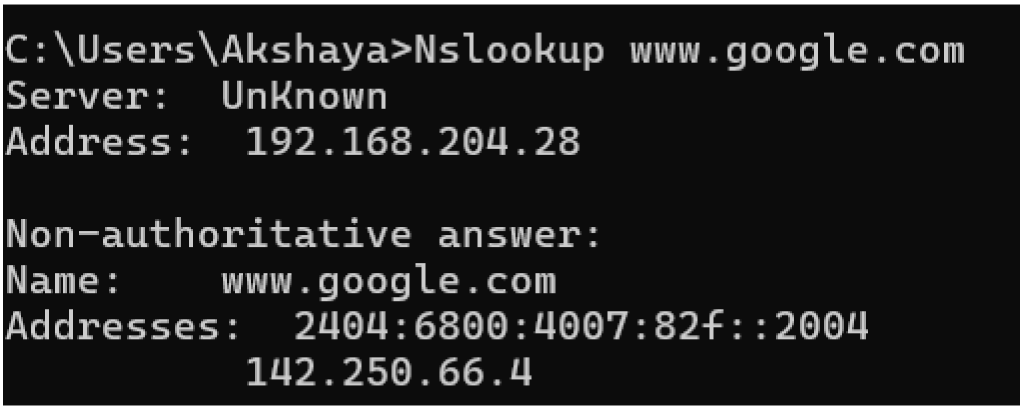
```
Microsoft Windows [Version 10.0.22631.5335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Akshaya>ipconfig

Windows IP Configuration


Ethernet adapter vEthernet (WSL):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8e68:b609:c19f:4bca%29
   IPv4 Address. . . . . . . . . . . : 172.24.16.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2401:4900:4ddd:2f40:736d:f3:34af:857f
   Temporary IPv6 Address. . . . . . : 2401:4900:4ddd:2f40:8c39:ba2d:b631:919c
   Link-local IPv6 Address . . . . . : fe80::61c1:15d3:7e88:3868%14
   IPv4 Address. . . . . . . . . . . : 192.168.204.87
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::e4ce:fcff:fe05:2af%14
                                       192.168.204.28

C:\Users\Akshaya>
```

2.Netstat



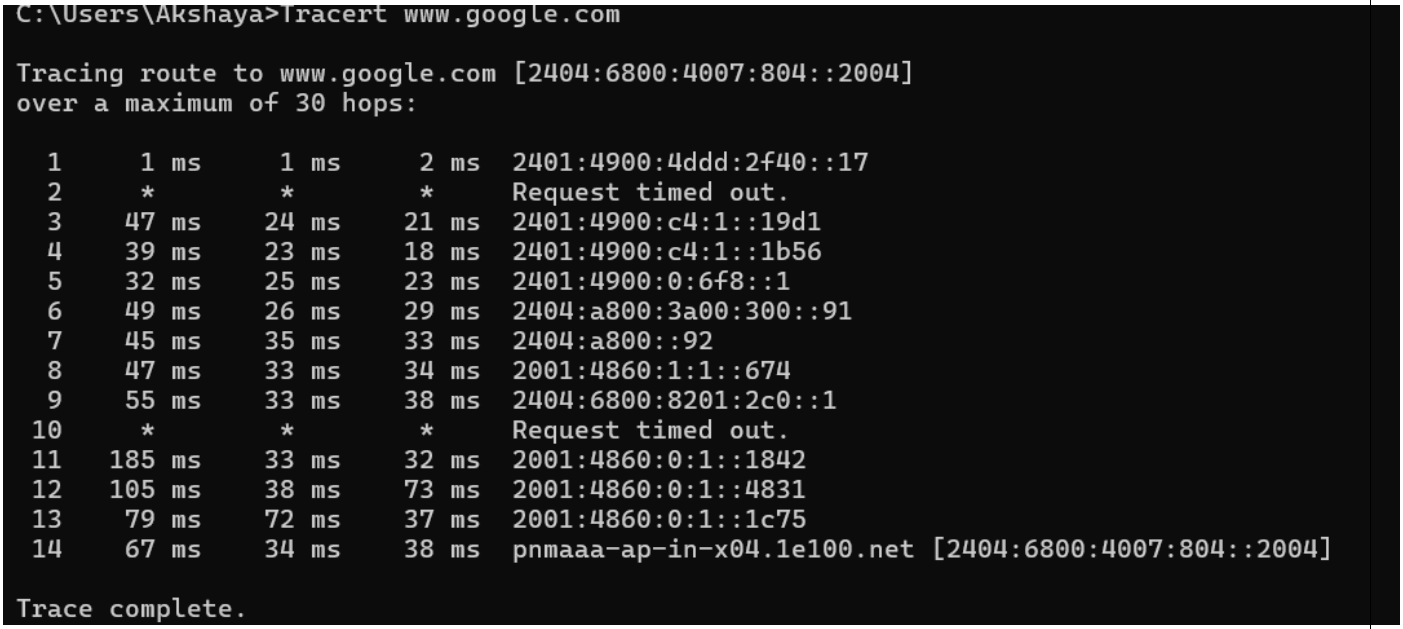
```
C:\Users\Akshaya>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.204.87:49427   4.213.25.240:https     ESTABLISHED
  TCP    192.168.204.87:64573   40.99.34.146:https     ESTABLISHED
  TCP    192.168.204.87:64574   40.99.34.146:https     ESTABLISHED
  TCP    192.168.204.87:64689   13.107.246.254:https   CLOSE_WAIT
  TCP    192.168.204.87:64703   52.98.86.162:https     TIME_WAIT
  TCP    192.168.204.87:64733   whatsapp-chatd-edge-shv-03-maa3:https  ESTABLISHED
  TCP    192.168.204.87:64948   relay-4d712b10:https   ESTABLISHED
  TCP    192.168.204.87:65054   hkg07s52-in-f3:https   TIME_WAIT
  TCP    192.168.204.87:65154   a23-205-118-49:https   ESTABLISHED
  TCP    192.168.204.87:65156   52.98.86.162:https     ESTABLISHED
  TCP    192.168.204.87:65165   a23-206-206-30:https   ESTABLISHED
  TCP    192.168.204.87:65166   a23-206-204-8:https    ESTABLISHED
  TCP    192.168.204.87:65167   a23-206-204-8:https    ESTABLISHED
  TCP    192.168.204.87:65168   a23-206-206-30:https   ESTABLISHED
  TCP    192.168.204.87:65169   a23-206-206-30:https   ESTABLISHED
  TCP    192.168.204.87:65170   a23-206-206-30:https   ESTABLISHED
  TCP    192.168.204.87:65171   a23-206-204-8:https    ESTABLISHED
  TCP    192.168.204.87:65173   13.107.3.254:https     ESTABLISHED
  TCP    192.168.204.87:65174   204.79.197.222:https   ESTABLISHED
  TCP    192.168.204.87:65180   13.107.4.254:https     ESTABLISHED
  TCP    192.168.204.87:65182   52.182.143.211:https   ESTABLISHED
  TCP    192.168.204.87:65183   52.182.143.211:https   ESTABLISHED
  TCP    192.168.204.87:65184   4.247.22.73:https      TIME_WAIT
  TCP    192.168.204.87:65185   13.107.137.11:https    ESTABLISHED
  TCP    192.168.204.87:65192   13.89.179.8:https      ESTABLISHED
  TCP    192.168.204.87:65193   52.104.98.39:https     ESTABLISHED
  TCP    192.168.204.87:65198   51.116.246.105:https   ESTABLISHED
  TCP    192.168.204.87:65204   4.247.22.73:https      TIME_WAIT
  TCP    192.168.204.87:65209   51.116.246.105:https   ESTABLISHED
  TCP    192.168.204.87:65210   4.247.22.73:https      TIME_WAIT
  TCP    192.168.204.87:65211   a23-12-238-186:https   ESTABLISHED
  TCP    192.168.204.87:65212   a23-12-238-186:https   ESTABLISHED
  TCP    [2401:4900:4ddd:2f40:8c39:ba2d:b631:919c]:64613  [2603:1040:a06:6::]:https  ESTABLISHED
  TCP    [2401:4900:4ddd:2f40:8c39:ba2d:b631:919c]:64619  [2603:1046:c06:803::2]:https  ESTABLISHED
  TCP    [2401:4900:4ddd:2f40:8c39:ba2d:b631:919c]:65172  [2620:1ec:bdf::254]:https  CLOSE_WAIT
  TCP    [2401:4900:4ddd:2f40:8c39:ba2d:b631:919c]:65186  g2600-140f-5400-0000-0000-0000-17dd-ee33:http  ESTABLISHED
```

3. Nslookup www.google.com



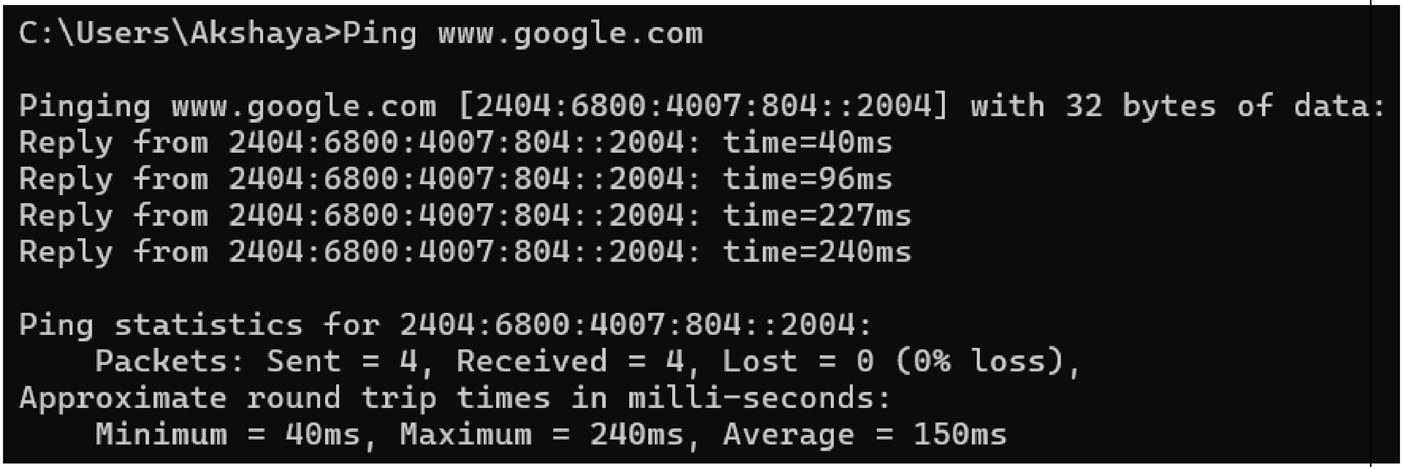
```
C:\Users\Akshaya>Nslookup www.google.com
Server:  UnKnown
Address:  192.168.204.28

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4007:82f::2004
          142.250.66.4
```

4.Tracert www.google.com



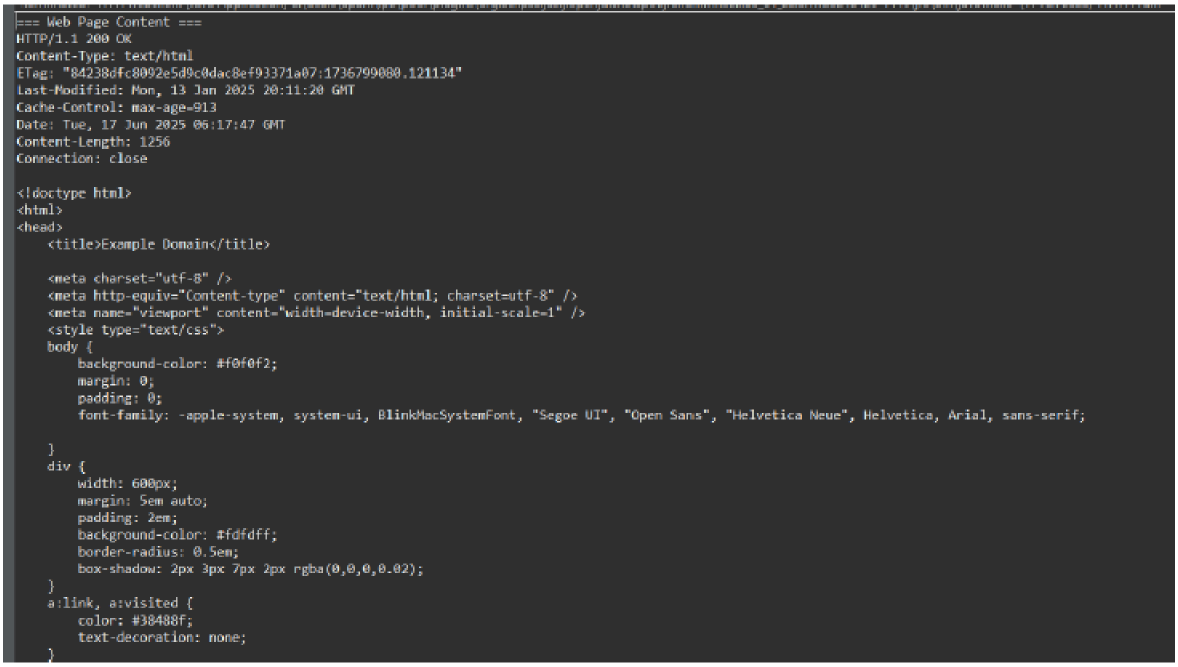
```
C:\Users\Akshaya>Tracert www.google.com

Tracing route to www.google.com [2404:6800:4007:804::2004]
over a maximum of 30 hops:

  1     1 ms     1 ms     2 ms  2401:4900:4ddd:2f40::17
  2     *        *        *     Request timed out.
  3    47 ms    24 ms    21 ms  2401:4900:c4:1::19d1
  4    39 ms    23 ms    18 ms  2401:4900:c4:1::1b56
  5    32 ms    25 ms    23 ms  2401:4900:0:6f8::1
  6    49 ms    26 ms    29 ms  2404:a800:3a00:300::91
  7    45 ms    35 ms    33 ms  2404:a800::92
  8    47 ms    33 ms    34 ms  2001:4860:1:1::674
  9    55 ms    33 ms    38 ms  2404:6800:8201:2c0::1
 10     *        *        *     Request timed out.
 11   185 ms    33 ms    32 ms  2001:4860:0:1::1842
 12   105 ms    38 ms    73 ms  2001:4860:0:1::4831
 13    79 ms    72 ms    37 ms  2001:4860:0:1::1c75
 14    67 ms    34 ms    38 ms  pnmaaa-ap-in-x04.1e100.net [2404:6800:4007:804::2004]

Trace complete.
```

5.Ping www.google.com

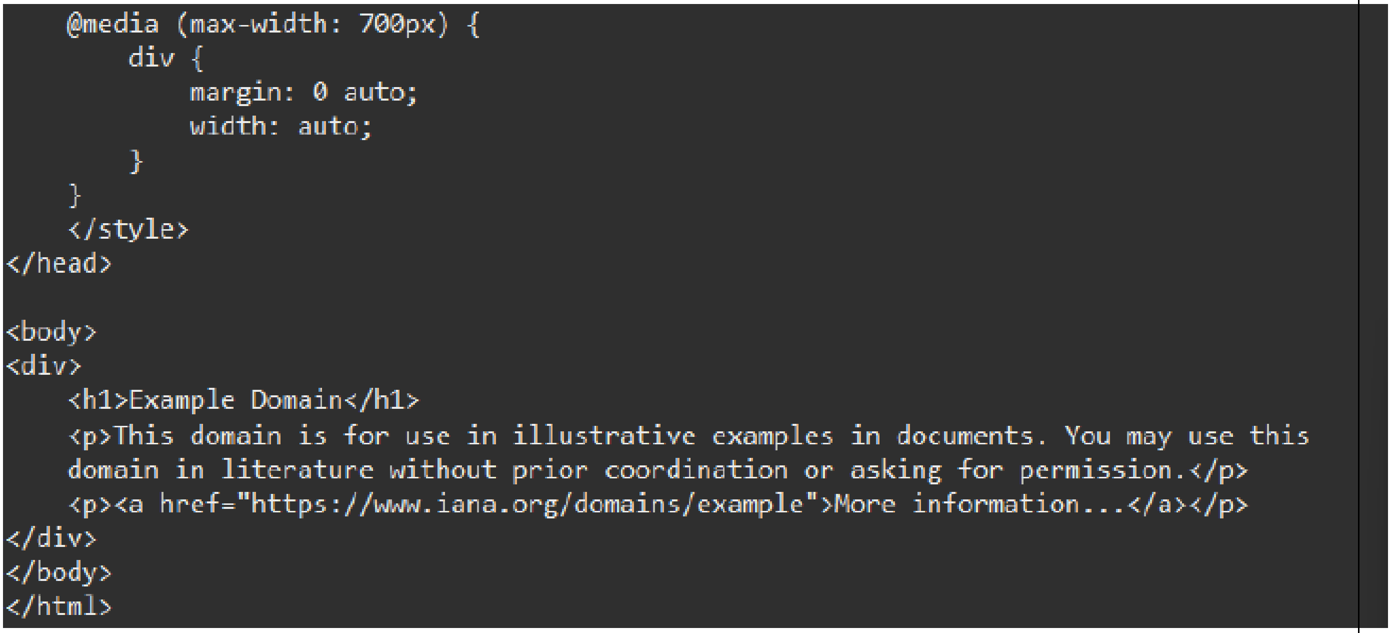


```
C:\Users\Akshaya>Ping www.google.com

Pinging www.google.com [2404:6800:4007:804::2004] with 32 bytes of data:
Reply from 2404:6800:4007:804::2004: time=40ms
Reply from 2404:6800:4007:804::2004: time=96ms
Reply from 2404:6800:4007:804::2004: time=227ms
Reply from 2404:6800:4007:804::2004: time=240ms

Ping statistics for 2404:6800:4007:804::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 40ms, Maximum = 240ms, Average = 150ms
```

OUTPUT :



```
=== Web Page Content ===
HTTP/1.1 200 OK
Content-Type: text/html
ETag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT
Cache-Control: max-age=913
Date: Tue, 17 Jun 2025 06:17:47 GMT
Content-Length: 1256
Connection: close

<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
```



```
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

3A .PROGRAM :

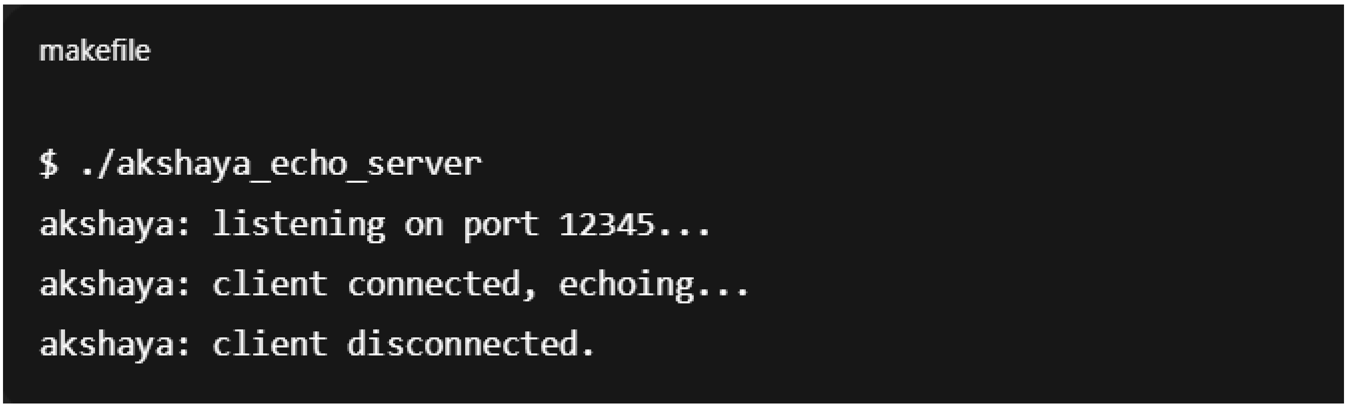
SERVER

```
// akshaya TCP echo server (iterative, single-client)
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define PORT 12345
#define BUFSIZE 1024
int main() {
  int listen_fd, conn_fd;
  struct sockaddr_in serv_addr;
  char buffer[BUFSIZE];
  socklen_t addrlen = sizeof(serv_addr);
  listen_fd = socket(AF_INET, SOCK_STREAM, 0);
  if (listen_fd < 0) { perror("socket"); exit(1); }
  memset(&serv_addr, 0, sizeof(serv_addr));
  serv_addr.sin_family = AF_INET;
  serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
  serv_addr.sin_port = htons(PORT);
  if (bind(listen_fd, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) < 0) {
    perror("bind"); exit(1);
  }
  if (listen(listen_fd, 5) < 0) { perror("listen"); exit(1); }
  printf("akshaya: listening on port %d...\n", PORT);
  while ((conn_fd = accept(listen_fd, (struct sockaddr*)NULL, NULL)) >= 0) {
    printf("akshaya: client connected, echoing...\n");
    ssize_t n;
    while ((n = read(conn_fd, buffer, BUFSIZE)) > 0) {
      write(conn_fd, buffer, n);
    }
    close(conn_fd);
    printf("akshaya: client disconnected.\n");
  }
  close(listen_fd);
  return 0;
}
```

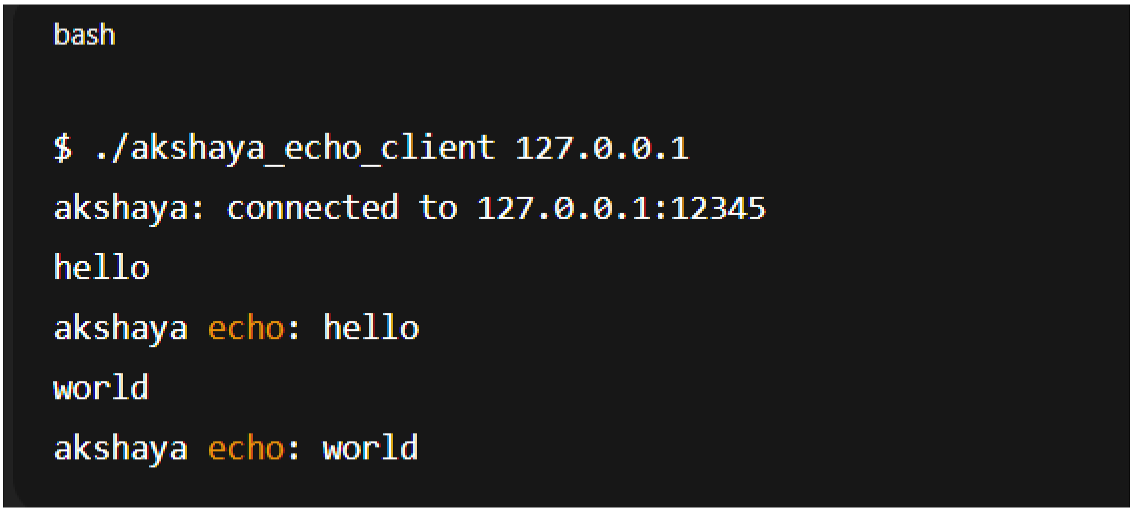
CLIENT

```
// akshaya TCP echo client
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define PORT 12345
#define BUFSIZE 1024
int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <server-ip>\n", argv[0]);
        return 1;
    }
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) { perror("socket"); return 1; }
    struct sockaddr_in serv_addr;
    memset(&serv_addr, 0, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);
    if (inet_pton(AF_INET, argv[1], &serv_addr.sin_addr) <= 0) {
        perror("inet_pton"); return 1;
    }
    if (connect(sockfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) < 0) {
        perror("connect"); return 1;
    }
    printf("akshaya: connected to %s:%d\n", argv[1], PORT);
    char buffer[BUFSIZE];
    while (fgets(buffer, BUFSIZE, stdin) != NULL) {
        write(sockfd, buffer, strlen(buffer));
        ssize_t n = read(sockfd, buffer, BUFSIZE);
        if (n <= 0) break;
        buffer[n] = '\0';
        printf("akshaya echo: %s", buffer);
    }
    close(sockfd);
    return 0;
}
```

OUTPUT :



```
makefile

$ ./akshaya_echo_server
akshaya: listening on port 12345...
akshaya: client connected, echoing...
akshaya: client disconnected.
```



```
bash

$ ./akshaya_echo_client 127.0.0.1
akshaya: connected to 127.0.0.1:12345
hello
akshaya echo: hello
world
akshaya echo: world
```

3B.PROGRAM :

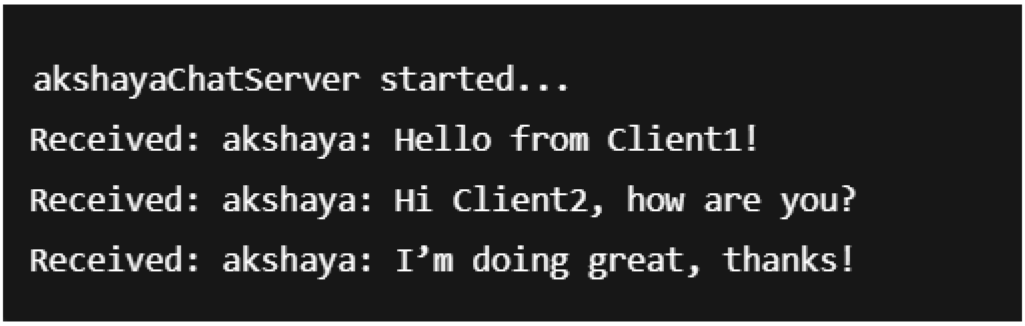
SERVER :

```
package akshaya.chat;
import java.io.*;
import java.net.*;
import java.util.*;
public class akshayaChatServer {
    private static final int PORT = 12345;
    private static Set<PrintWriter> clientWriters = new HashSet<>();
    public static void main(String[] args) {
        System.out.println("akshayaChatServer started...");
        try (ServerSocket serverSocket = new ServerSocket(PORT)) {
            while (true) {
                Socket socket = serverSocket.accept();
                new akshayaClientHandler(socket).start();
            }
        } catch (IOException e) {
            System.out.println("Error starting akshayaChatServer: " + e.getMessage());
        }
    }
    private static class akshayaClientHandler extends Thread {
        private Socket socket;
        private PrintWriter out;
        public akshayaClientHandler(Socket socket) {
            this.socket = socket;
        }
        public void run() {
            try (
                InputStreamReader isr = new InputStreamReader(socket.getInputStream());
                BufferedReader in = new BufferedReader(isr);
            ) {
                out = new PrintWriter(socket.getOutputStream(), true);
                clientWriters.add(out);
                String message;
                while ((message = in.readLine()) != null) {
                    System.out.println("Received: " + message);
                    for (PrintWriter writer : clientWriters) {
                        writer.println(message);
                    }
                }
            } catch (IOException e) {
                System.out.println("Connection error: " + e.getMessage());
            } finally {
                if (out != null) {
                    clientWriters.remove(out);
                }
                try { socket.close(); } catch (IOException ignored) {}
            }
        }
```

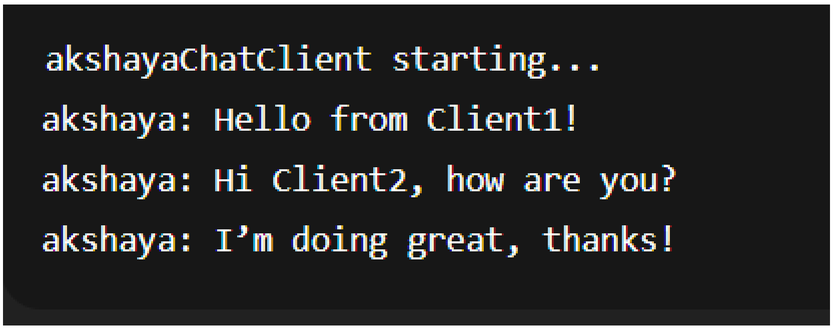
CLIENT :

```
package akshaya.chat;
import java.io.*;
import java.net.*;
public class akshayaChatClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 12345;
    public static void main(String[] args) {
        System.out.println("akshayaChatClient starting...");
        try (
            Socket socket = new Socket(SERVER_ADDRESS, SERVER_PORT);
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
            BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
        ) {
            // Thread for receiving messages from server
            new Thread(() -> {
                try {
                    String serverMsg;
                    while ((serverMsg = in.readLine()) != null) {
                        System.out.println(serverMsg);
                    }
                } catch (IOException e) {
                    System.out.println("Disconnected from server.");
                }
            }).start();
            // Sending user input to server
            String input;
            while ((input = userInput.readLine()) != null) {
                out.println("akshaya: " + input);
            }
        } catch (IOException e) {
            System.err.println("Cannot connect to akshayaChatServer: " + e.getMessage());
        }
    }
}
```

OUTPUT :



```
akshayaChatServer started...
Received: akshaya: Hello from Client1!
Received: akshaya: Hi Client2, how are you?
Received: akshaya: I’m doing great, thanks!
```



```
akshayaChatClient starting...
akshaya: Hello from Client1!
akshaya: Hi Client2, how are you?
akshaya: I’m doing great, thanks!
```

4.PROGRAM :

SERVER :

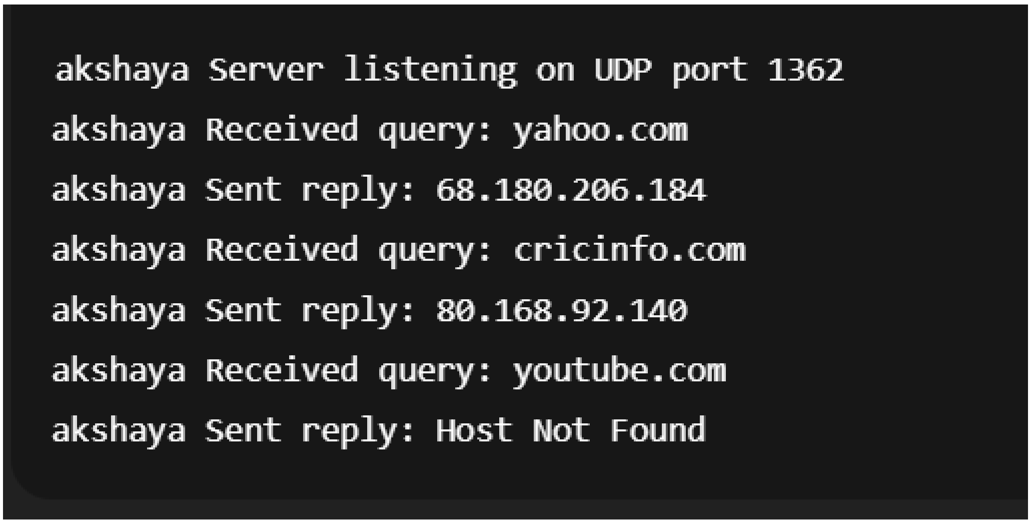
```
package akshaya.dns;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class akshayaUDPDNSServer {
  private static final int PORT = 1362;
  public static void main(String[] args) {
    String[] hosts = {"yahoo.com", "gmail.com", "cricinfo.com", "facebook.com"};
    String[] ips   = {"68.180.206.184", "209.85.148.19", "80.168.92.140", "69.63.189.16"};
    System.out.println("akshaya Server listening on UDP port " + PORT);
    try (DatagramSocket serverSocket = new DatagramSocket(PORT)) {
      byte[] buffer = new byte[1024];
      while (true) {
        DatagramPacket request = new DatagramPacket(buffer, buffer.length);
        serverSocket.receive(request);
        String query = new String(request.getData(), 0, request.getLength()).trim();
        System.out.println("akshaya Received query: " + query);
        String replyStr = "Host Not Found";
        for (int i = 0; i < hosts.length; i++) {
          if (hosts[i].equalsIgnoreCase(query)) {
            replyStr = ips[i];
            break;
          }
        }
        byte[] sendData = replyStr.getBytes();
        DatagramPacket reply = new DatagramPacket(
          sendData, sendData.length,
          request.getAddress(), request.getPort()
        );
        serverSocket.send(reply);
        System.out.println("akshaya Sent reply: " + replyStr);
      }
```

```
        } catch (IOException e) {
            System.err.println("akshaya Server error: " + e.getMessage());
        }
    }
}
```

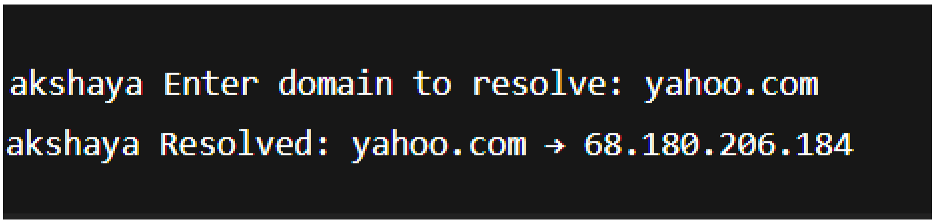
CLIENT :

```
package akshaya.dns;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class akshayaUDPDNSClient {
    private static final int PORT = 1362;
    private static final String SERVER = "localhost"
    public static void main(String[] args) {
        try (DatagramSocket clientSocket = new DatagramSocket();
             BufferedReader userIn = new BufferedReader(new InputStreamReader(System.in))) {
            System.out.print("akshaya Enter domain to resolve: ");
            String domain = userIn.readLine().trim();
            byte[] sendData = domain.getBytes();
            InetAddress addr = InetAddress.getByName(SERVER);
            DatagramPacket request = new DatagramPacket(sendData, sendData.length, addr, PORT);
            clientSocket.send(request);
            byte[] receiveData = new byte[1024];
            DatagramPacket reply = new DatagramPacket(receiveData, receiveData.length);
            clientSocket.receive(reply);
            String result = new String(reply.getData(), 0, reply.getLength());
            System.out.println("akshaya Resolved: " + domain + " → " + result);
        } catch (IOException e) {
            System.err.println("akshaya Client error: " + e.getMessage());
        }
    }
}
```

OUTPUT :



```
akshaya Server listening on UDP port 1362
akshaya Received query: yahoo.com
akshaya Sent reply: 68.180.206.184
akshaya Received query: cricinfo.com
akshaya Sent reply: 80.168.92.140
akshaya Received query: youtube.com
akshaya Sent reply: Host Not Found
```



```
akshaya Enter domain to resolve: yahoo.com
akshaya Resolved: yahoo.com → 68.180.206.184
```