Practical 12

AIM:

a) Implement echo client Server using TCP/UDP.

client:

```
import socket
import time
def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, SOCKET_
    SOCK_ROUTE)
        try:
            s.sendto(b'Hello', (host, port))
        except.timeout
            Print("Request timed out");
    if name_=="_main_":
        Ping.Server().
```

Server

```
import server
    def start_server host = "127.0.0.1', port=123,45):
        with socket.socket(socket.AF.NET, SOCKET.SOCK_DGRAM)
        s.bind((host, port))
        Print(f"UDP source running on short")
    while True:
        dataaddr : S.recvfrom(1024)
    Print+("Received menage")
if __name__=="__main__":
    start.server()
```

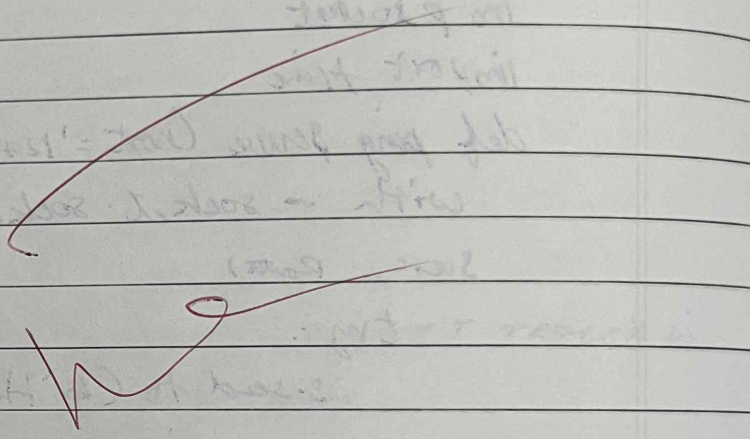
O/P:

```
python receive.py
UDP server running on 127.0.0.1 : 12345
Received menage from (127.0.0.1, 59220)
```

Python client.py
Received from sever : Hello client.

Result.

Thus the program was successfully executed.



AIM:

Implement chat 'client' server using TCP / UDP sockets.

Chat server.py

```
import socket
def receiver():
    Port = 12345
    host = '127.0.0.1'
    with socket.socket(socket.AF_INET, SOCK_PROGRAM) as s:
        s.bind((host, port))
    while(True):
        d,addd = s.recvfrom('1024')
        Print("Client")
        a = input("Enter Reply")
        s.sendto(a.encode(), addd)
        if (a == 'end')
            break
            exit
        recv()
```

receiver.py

```
import socket
import time
def receiver(a):
    host = "127.0.0.1".
    while (TRUE):
        a = input("Enter message")
        if (a == 'end')
            receiver(0)
            break
    else
        receiver(a)
```

O/P:

```
Python ./check.py
client {'hi'}
client {'How are you'}
  Enter reply I'm fine

Python ./recer.py
Enter message :hi
{"Hello"}
Enter message How are you
{"I am fine"}
```

Result:

Thus the client server using UDP/TCP is executed and output got successfully.