Echo client-server using TCP/UDP sockets

Ex no: 12a
Date : 15/10/24

AIM :
Implement echo client server using TCP/UDP sockets.

Client:

```
import socket
import time
def ping-server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET,
        socket.SOCK_DGRAM) as s:
        try:
            s.sendto(b'Hello', (host, port))
        except s.timeout:
            print("Request timed out")

if __name__ == "main":
    ping-server()
```

Server:

```
import socket
def start_server(host='127.0.0.1', port=12345)
with socket.socket(socket.AF_INET, socke.SOC_DGRA) as s
s.bind((host, porty)
print(f"UDP server running on {hosty} {porty})
while True:
    data, addr : s.recvfrom(1024)
    print(f"Received message from {addry} {data
                                    decode}")}

if __name__ = "main"
    start.server()
```

O/P : python server.py

```
UDP server running on 127.0.0.1 12345
Received message from (127.0.0.1, 12345)
                                Hello
```

python client.py

```
Received reply from server: Hello, client.
```

b) Implement chat client server using UDP / TCP:

chat server.py

```
import socket
def recvr1():
    port=12345
    host='127.0.0.1'
with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s
    s.bind({host,port})
    while (True)
        d,add=s.recvfrom(1024)
        print("client",{d.decode()})
        a=input("Enter Reply")
        s.sendto(a.encode(),add)
    if(a=='end')
        break
        exit

recvr()
```

recvr2.py

```
import socket
import time
def recvr2(a):
    host='127.0.0.1'
    port=12345
with socket.socket(socket.AF_INET,socket.SOCK_DGRAM)as s:
        s.sendto(a.encode(),(host,port))
        d,saddr=s.recvfrom(1024)
        print({d.decode()})
while(True):
        a=input("Enter Message");
        if(a=='end'):
            recvr2(a)
            break
        else:
            recvr2(a)
```

O/P:

O/P: python chat-serv.py

client {'hi'}

Enter Reply hello

Enter reply Im fine

Client {" How are you"}

python recv.py

Enter message hi

{' hello'}

Enter message How are you

{' I m fine'}

Enter message

RESULT:

Thus the program is executed and the output is verified