Problem No: 03
Problem Name: Implement and establish “Socket Server and Multi-Threaded Client Communication in Python”

Objective:
To learn how to build a concurrent network application where a server handles multiple client connections at the same time using threads.

Theory:
A socket provides two-way communication between devices. In a multi-threaded client–server model, the server creates a new thread for each client connection, allowing parallel communication without blocking. This improves responsiveness and scalability.

Key Components:
- Python socket programming
- Thread creation and management
- TCP server–client architecture
- Concurrent client handling

Application:
Used in chat systems, real-time collaboration tools, multiplayer games, notification services, and any multi-user network-based system.

Implementation in Python:
Server Code (server1002.py)
```
import socket,threading
HOST="127.0.0.1";PORT=1002
def handle(c,a):
   print(f"[CONNECTED]{a}connected.")
   while True:
      try:
         d=c.recv(1024).decode()
         if not d:break
         c.send(("Special word received: Monowar detected!"if d.lower()=="monowar"else f"Echo from server:{d}").encode())
      except:break
   print(f"[DISCONNECTED]{a}disconnected.");c.close()
s=socket.socket();s.bind((HOST,PORT));s.listen()
print(f"[SERVER RUNNING]Listening on port{PORT}...")
while True:
   c,a=s.accept()
   threading.Thread(target=handle,args=(c,a)).start()
   print(f"[ACTIVE CONNECTIONS]{threading.active_count()-1}")
```

Client Code (client1002.py)
```
import socket
HOST="127.0.0.1"
PORT=1002
c=socket.socket()
c.connect((HOST,PORT))
```

```
print("Connected to server.")
while True:
    m=input("chiti deo: ")
    if m.lower()=="shishir":break
    c.send(m.encode())
    print("Server:",c.recv(1024).decode())
c.close()
```

Result:
The server successfully handles multiple clients at the same time, each exchanging messages independently. The system remains responsive even with concurrent users.

Sample Output:

Server Console:
[SERVER RUNNING]Listening on port1002...
[CONNECTED]('127.0.0.1', 65158)connected.
[ACTIVE CONNECTIONS]1
[DISCONNECTED]('127.0.0.1', 65158)disconnected.

Client Console:
Connected to server.
chiti deo: monowar
Server: Special word received: Monowar detected!
chiti deo: exit
Server: Echo from server:exit
chiti deo: shishir

Discussion :
The experiment shows how threading resolves blocking behavior in basic sockets and enables parallel message exchanges. It highlights performance benefits and potential challenges such as synchronization and resource management.

Conclusion :
Multi-threading allows the server to manage multiple clients concurrently, making the communication system more efficient and scalable. This lab provides the foundation for building advanced multi-user network applications.