1. **Write a TCP socket program to establish connection between client and server. The client program will send an input value n to the server and the server program will return the sum of the square of first n natural numbers. Client will display the value send by server. The communication between client and server will continue until client send 'Quit' message to the server.**

- **Code:**

- **server1.py**

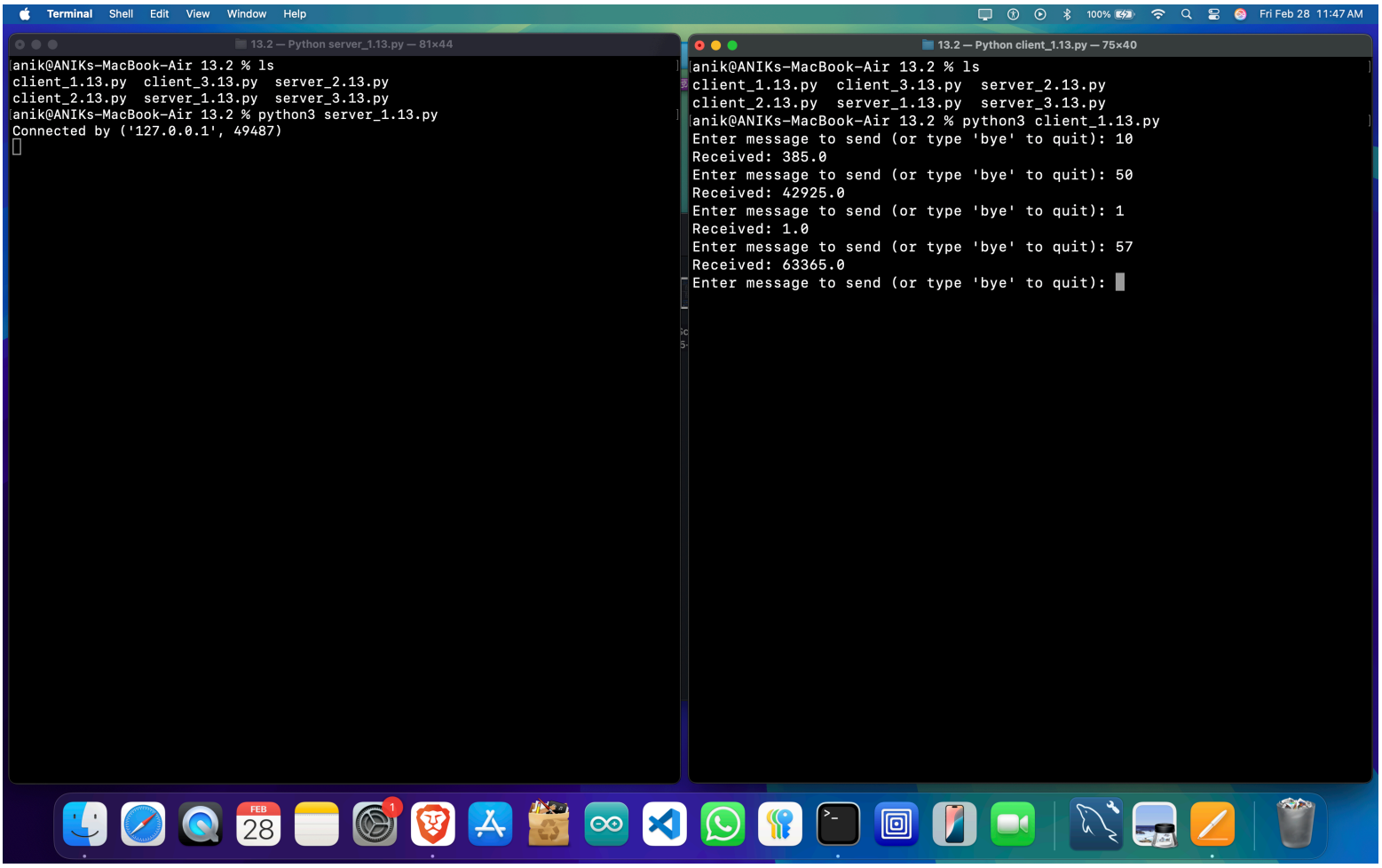
```
import socket

def server1():
    host = '127.0.0.1'
    port = 8000

    s_socket = socket.socket()
    s_socket.bind((host, port))
    s_socket.listen(1)
    print("Server is
    listening...")

    conn, addr = s_socket.accept()
    print("Server is connected by", addr)

    while True:
        val =
        conn.recv(1024).decode() if
        val.lower() == 'quit':
            break

        try:
            n = int(val)
            result = (n*(n+1)*(2*n+1))/6
            conn.sendall(str(result).encode())
            print('Result is sent successfully')
```

```
    s_socket.close()

server1()
```

- **client1.py**

```
import socket

def client1():
    host = '127.0.0.1'
    port = 8000

    c_socket = socket.socket()
    c_socket.connect((host, port))

    while True:
        n = input("Enter the input n (Quit to exit) :")
        c_socket.sendall(n.encode())

        if n.lower() == 'quit':
            break

        resp = c_socket.recv(1024).decode()
        print("Sum of Square of n natural number:", resp)

    c_socket.close()

client1()
```

- **Output:**


Terminal Shell Edit View Window Help
Fri Feb 28 11:47 AM
13.2 — Python server_1.13.py — 81×44
anik@ANIKs-MacBook-Air 13.2 % ls
client_1.13.py client_3.13.py server_2.13.py
client_2.13.py server_1.13.py server_3.13.py
anik@ANIKs-MacBook-Air 13.2 % python3 server_1.13.py
Connected by ('127.0.0.1', 49487)
13.2 — Python client_1.13.py — 75×40
anik@ANIKs-MacBook-Air 13.2 % ls
client_1.13.py client_3.13.py server_2.13.py
client_2.13.py server_1.13.py server_3.13.py
anik@ANIKs-MacBook-Air 13.2 % python3 client_1.13.py
Enter message to send (or type 'bye' to quit): 10
Received: 385.0
Enter message to send (or type 'bye' to quit): 50
Received: 42925.0
Enter message to send (or type 'bye' to quit): 1
Received: 1.0
Enter message to send (or type 'bye' to quit): 57
Received: 63365.0
Enter message to send (or type 'bye' to quit):

2. **To write a TCP socket program (in C/C++/Java/Python) to establish connection between client and server. The client program will send a set of binary values to the server and the server program will return the number of 1s present in the data received. Client will display the value send by server. The communication between client and server will continue until client send 'Quit' message to the server.**

   - **Code:**

   - **server2.py**

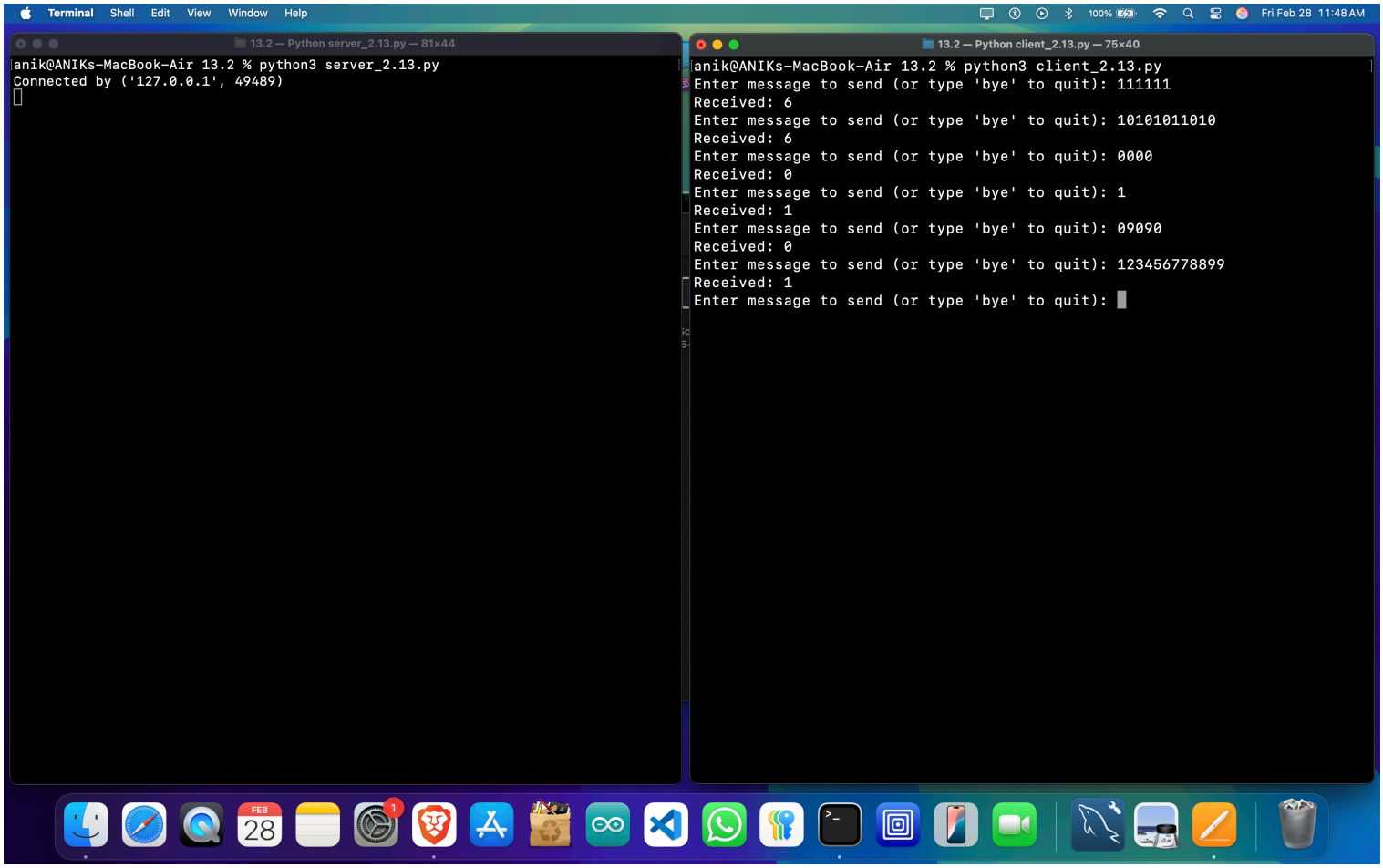
```
import socket

def server2():
    host = '127.0.0.1'
    port = 8001

    s_socket = socket.socket()
    s_socket.bind((host, port))
    s_socket.listen(1)
    print("Server is
    listening...")

    conn, addr = s_socket.accept()
    print("Server is connected by", addr)

    while True:
        val =
        conn.recv(1024).decode() if
        val.lower() == 'quit':
            break

        try:
            count=0
            for i in
                range(len(val)): if
                val[i]=='1':
                    count+=1
```

```
    conn.close()
    s_socket.close()

server2()

s_socket.close()
```

- **client2.py**

```
import socket

def client2():
    host = '127.0.0.1'
    port = 8001

    c_socket = socket.socket()
    c_socket.connect((host, port))

    while True:
        val = input("Enter a  set of binary values :")
        c_socket.sendall(val.encode())

        if val.lower() == 'quit':
            break

        resp = c_socket.recv(1024).decode()
        print("The number of 1s :", resp)

    c_socket.close()

client2()
```

- **Output:**


Terminal Shell Edit View Window Help
Fri Feb 28 11:48 AM
13.2 — Python server_2.13.py — 81×44
anik@ANIKs-MacBook-Air 13.2 % python3 server_2.13.py
Connected by ('127.0.0.1', 49489)
13.2 — Python client_2.13.py — 75×40
anik@ANIKs-MacBook-Air 13.2 % python3 client_2.13.py
Enter message to send (or type 'bye' to quit): 111111
Received: 6
Enter message to send (or type 'bye' to quit): 10101011010
Received: 6
Enter message to send (or type 'bye' to quit): 0000
Received: 0
Enter message to send (or type 'bye' to quit): 1
Received: 1
Enter message to send (or type 'bye' to quit): 09090
Received: 0
Enter message to send (or type 'bye' to quit): 123456778899
Received: 1
Enter message to send (or type 'bye' to quit):

3. **To write a Write a TCP socket program (in C/C++/Java/Python) to establish connection between client and server. The client program will send a postfix expression to the server and the server program will return the result of the input expression. Server program will use a stack to evaluate the postfix expression. Client will display the value send by server. The communication between client and server will continue until client send 'Quit' message to the server.**

- **Code:**

- **server3.py**

```
import socket

def server3():
    host = '127.0.0.1'
    port = 8002

    s_socket = socket.socket()
    s_socket.bind((host, port))
    s_socket.listen(1)
    print("Server is
    listening...")

    conn, addr = s_socket.accept()
    print("Server is connected by", addr)

    while True:
        exp =
        conn.recv(1024).decode() if
        exp.lower() == 'quit':
            break

        try:
            s=[]
            op={'+','-','*','/'}

            ts=exp.split(",")
```

```
                if len(s)<2:
                    print('Error'
                    ) break

                a=s.pop()
                b=s.pop()

                if t=='+':
                    s.append(b+a)
                elif t=='-':
                    s.append(b-a)
                elif t=='*':
                    s.append(b*a)
                elif t=='/':
                    if a!=0:
                        s.append(b/a)
            else:
                print('Error')
                break


        res=s[0]

        conn.sendall(str(res).encode())
        print('Result is sent successfully')
    except ValueError:
        conn.sendall(b"Invalid input")

conn.close()
s_socket.close()
```

- **client3.py**

```
import socket

def client3():
    host = '127.0.0.1'
    port = 8002

    c_socket = socket.socket()
    c_socket.connect((host, port))
```

```python
while True:
    exp = input("Enter the postfix expression (seperated by ',' ):")
    c_socket.sendall(exp.encode())

    if exp.lower() == 'quit':
        break

    resp = c_socket.recv(1024).decode()
    print("The Evaluated value of Postfix Expression:", resp)

c_socket.close()
```

- **Output:**