

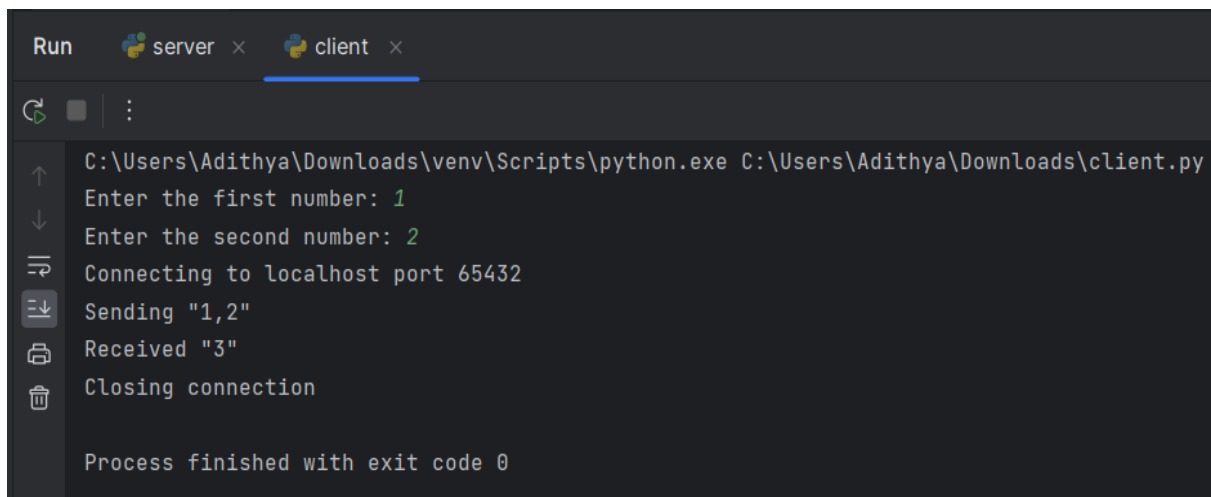
SERVER

```
clientdhcp.py  serverdns.py  clientdns.py  serverdhcp.py  server.py  client.py
1  import socket
2  1 usage
3  def start_server():
4      server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5      server_address = ('localhost', 65432)
6      print(f'Starting server on {server_address[0]} port {server_address[1]}')
7      server_socket.bind(server_address)
8      server_socket.listen(1)
9      while True:
10         print('Waiting for a connection...')
11         connection, client_address = server_socket.accept()
12         try:
13             print(f'Connection from {client_address}')
14             while True:
15                 data = connection.recv(1024)
16                 if data:
17                     numbers = data.decode('utf-8').split(',')
18                     if len(numbers) == 2:
19                         num1 = int(numbers[0])
20                         num2 = int(numbers[1])
21                         result = num1 + num2
22                         response = str(result)
23                         connection.sendall(response.encode('utf-8'))
24                     else:
25                         connection.sendall(b'Invalid input')
26                 else:
27                     break
28             finally:
29                 connection.close()
30 if __name__ == '__main__':
31     start_server()
```

CLIENT

```
clientdhcp.py  serverdns.py  clientdns.py  serverdhcp.py  server.py  client.py
1  import socket
2
3  1 usage
4  def send_request(num1, num2):
5      client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6      server_address = ('localhost', 65432)
7      print(f'Connecting to {server_address[0]} port {server_address[1]}')
8      client_socket.connect(server_address)
9
10     try:
11         message = f'{num1},{num2}'
12         print(f'Sending "{message}"')
13         client_socket.sendall(message.encode('utf-8'))
14         response = client_socket.recv(1024)
15         print(f'Received "{response.decode("utf-8")}"')
16     finally:
17         print('Closing connection')
18         client_socket.close()
19
20 if __name__ == '__main__':
21     num1 = int(input("Enter the first number: "))
22     num2 = int(input("Enter the second number: "))
23     send_request(num1, num2)
```

OUTPUT



The screenshot shows a code editor with two tabs: 'server' and 'client'. The 'client' tab is active. The output of the script is displayed in a terminal window. The output shows the client connecting to the server, sending a message, and receiving a response.

```
Run server x client x
C:\Users\Adithya\Downloads\venv\Scripts\python.exe C:\Users\Adithya\Downloads\client.py
Enter the first number: 1
Enter the second number: 2
Connecting to localhost port 65432
Sending "1,2"
Received "3"
Closing connection

Process finished with exit code 0
```