Calculator

Server:

```python
import socket

s = socket.socket()
s.bind(("localhost" , 9999))
s.listen(5)
print("listening on port 9999")
while True:
    c,c_addr = s.accept()

    c.send("Enter two numbers and operator to perform calculation in format
(num1,num2,op)".encode())
    print("Waiting for client to respond\n")

    data = c.recv(1024).decode()

    #split data in num1, num2, and operator current format (num1,num2,op)
    num1,  num2 , op = data.split(",")

    #typecast num1 and num2

    num1 , num2 = float(num1),float(num2)
    ans = 0
    print("received numbers: ",num1,num2)

    if op == '+':
        ans = num1 + num2

    elif op == '-':
        ans = num1 - num2

    elif op == '*':
        ans = num1 * num2

    elif op == '/':
        ans = num1 / num2
    else:
        result = "Invalid operator"

    #text file:
    with open("ans.txt","a") as file:
        file.write(f"{num1} {op} {num2} = {ans}")
```

```
    # # Send file to client
    # with open("result.txt", "rb") as file:
    #     file_data = file.read()
    #     c.sendall(file_data)  # Send entire file contents

    c.send(str(ans).encode())


    c.close()

   #to send and receive files, we will open them in binary mode (rb,wb)
```

client:

```
import socket

c = socket.socket()

c.connect(("localhost",9999))
print("Client connected to port 9999")

s = c.recv(1024).decode()
print(s);

num1 = input("Enter num1: ")
num2 = input("Enter num2: ")
op = input("Enter operator (+, -, *, /): ")

data = f"{num1},{num2},{op}" #2,2,+
print(data)
c.send(data.encode())

ans = c.recv(1024).decode()
print("Ans: ", ans)
# # Receive file from server
# file_data = c.recv(1024)

# # Save received file
# with open("received_result.txt", "wb") as file:
#     file.write(file_data)

# print("Result file received and saved as 'received_result.txt'.")
c.close()
```

Area & parameters + log file

Server:

```
import socket
import time

s = socket.socket()

s.bind(("localhost",9999))
s.listen(5)
print("server listening...")

area = 0
para = 0

while True:
    start = time.time()     #time started
    c,c_addr = s.accept()
    print("connected with client...")

    c.send("square or rectangle?".encode())
    shape = c.recv(1024).decode()

    c.send("Area or parameter?".encode())
    operation = c.recv(1024).decode()

    if shape == 'square' and operation == 'area':
        c.send("Enter square side size: ".encode())
        side = c.recv(1024).decode()
        side = int(side)
        area = side*side
        c.send(str(area).encode())

    elif shape == 'square' and operation == 'parameter':
        c.send("Enter square side size: ".encode())
        side = c.recv(1024).decode()
        side = int(side)
        para = 4*side
        c.send(str(para).encode())

    elif shape == 'rectangle' and operation == 'area':
        c.send("Enter length and breadth of rectangle in format (l,b) ".encode())
        data = c.recv(1024).decode()
        l,b = data.split(",")
```

```
        l = float(l)
        b = float(b)
        area = l*b
        c.send(str(area).encode())

    elif shape == 'rectangle' and operation == 'parameter':
        c.send("Enter length and breadth of rectangle: ".encode())
        data = c.recv(1024).decode()
        l,b = data.split(",")
        l = float(l)
        b = float(b)
        para = 2*(l+b)
        c.send(str(para).encode())

    end = time.time()
    duration = end - start

    with open("file.txt","a") as file:
        file.write(f"Duration of connectivity: {duration} \n")
        file.write(f"{shape} & {operation}: ")
        if operation == 'parameter':
            file.write(f"{para} ")
        else:
            file.write(f"{area}")

    print("ans sent successfully")
    c.close()
```

client:

```
import socket

c = socket.socket()

c.connect(("localhost",9999))

query1 = c.recv(1024).decode() #shape
shape = input("Enter shape: ")
c.send(shape.encode())

query2 = c.recv(1024).decode() #area or parameter
op = input("Area or Parameter: ")
c.send(op.encode())
```

```
data = c.recv(1024).decode()

if shape == 'square':
    l = input(f"enter side size of {shape}: ")
    c.send(l.encode())
else:
    l = input(f"enter length of {shape}: ")
    b = input(f"enter breadth of {shape}: ")
    ans = f"{l},{b}"
    c.send(ans.encode())

final = c.recv(1024).decode()
print("Ans: ",final)
```

Factorial + Fib + isPrime

Server:

```
import socket
import time

def factorial(num):
    if num <= 1:
        return 1
    else:
        return num*factorial(num-1)


def isPrime(num):
    for i in range(2,num+1):
        if num%i == 0:
            return False    # If divisible, it's not prime

    return True

def fib(num):
    if num <= 0:
        return 0
    elif num == 1:
        return 1
    else:
```

```
        return fib(num-1) + fib(num-2)

s = socket.socket()
s.bind(("localhost",9999))
s.listen(10)
print("Server listening on port 9999")

while True:
    c,c_addr = s.accept()
    start = time.time()     #timer
    print("connection established")

    num = c.recv(1024).decode()   #string
    num = int(num)

    print("Checking if Factorial of number...")
    fact = factorial(num)
    print("Factorial: ", fact)

    print("Checking if number is prime...")
    prime = isPrime(num)
    print(f"Is number prime: {prime} ")

    print("Fibonacci Sequence...")
    fibo = fib(num)
    print(f"Fib Seq: {fibo} ")

    end = time.time()
    dur = end-start

    with open("file.txt","a") as file:
        file.write(f'Number from client: {num} \nFactorial of {num}: {fact} \nIs
{num} Prime? {prime}\nFibonacci: {fibo} \nDuration Of connection: {dur}\n')

    c.close()
```

Client:

```
import socket

c = socket.socket()

c.connect(('localhost',9999))
print("connection established")
```

```
num = input("Enter number for finding Factorial, Prime Check , & Fibonacci
Sequence: ")

c.send(num.encode())
print(f"{num} sent to Server. Go to server's terminal to find out ans or find
them in file.txt. ")

c.close();
```

String Reversal + Palindrome

Server:

```
import socket
import time

s = socket.socket()
s.bind(('localhost',9999))
s.listen(10)
print("Server listening on port 9999")

while True:
    c,c_addr = s.accept()
    print("connection established")

    start = time.time()     #timer

    #recv the string to reverse and palindrome check
    palindrome = False
    str = c.recv(1024).decode()
    print("string to check: ",str)

    str_rev = str[::-1]
    print("reversed string: ",str_rev)
    print("Server checking for palindrome...")

    if str == str_rev:
        print("It is a palindrome.\n")
        palindrome = True
    else:
        print("It is NOT a palindrome.\n")

    end = time.time()
    dur = end - start
```

```
    print("Duration for connectivity: ",dur)

    with open("file.txt","a") as file:
        file.write(f"String recv from client: {str} \n String after reversal :
{str_rev} \n Is String Palindrome? {palindrome} \n\n")

    c.close()
```

Client:

```
import socket

c = socket.socket()

c.connect(("localhost",9999))
print("Client Connected with server")

str = input("Enter the string: ")
print(f'String: {str}')

c.send(str.encode())
print("String sent to server")

c.close()
```