'''Name - Mane Shivraj Pandurang.
Class - BE Artificial Intelligence and Data Science.
Roll No. - 37
Practical No. 01- Design a distributed application using RPC for remote computation where client submits an integer value to the server and server calculates factorial and returns the result to the client program.'''

Source Code –

Client.py

Factclient.py > ...

```python
import xmlrpc.client

# Create an XML-RPC client
with xmlrpc.client.ServerProxy("http://localhost:8000/RPC2") as proxy:
    try:
        # Prompt the user to enter a number
        input_value_str = input("Enter the number: ")
        input_value = int(input_value_str)

        # Call the server's calculate_factorial method
        result = proxy.calculate_factorial(input_value)
        print(f"Factorial of {input_value} is: {result}")
    except Exception as e:
        print(f"Error: {e}")

```

Output -
Enter the number: 12

Factorial of 12 is: 479001600

Server.py

Factserver.py > ...

```python
from xmlrpc.server import SimpleXMLRPCServer
from xmlrpc.server import SimpleXMLRPCRequestHandler

class FactorialServer:
    def calculate_factorial(self, n):
        if n < 0:
            raise ValueError("Input must be a non-negative integer.")
        result = 1
        for i in range(1, n + 1):
            result *= i
        return result

# Restrict to a particular path
class RequestHandler(SimpleXMLRPCRequestHandler):
    rpc_paths = ('/RPC2',)

# Create server
with SimpleXMLRPCServer(('localhost', 8000), requestHandler=RequestHandler) as server:
    server.register_introspection_functions()
    server.register_instance(FactorialServer())
    print("FactorialServer is ready to accept requests.")
    server.serve_forever()

```

Output - FactorialServer is ready to accept requests.