

Ex No : 11 B**Date :7/11/2024****Develop a simple calculator using XMLRPC****AIM : Develop a simple calculator using XMLRPC****XML RPC PROGRAM- SERVER SIDE:**

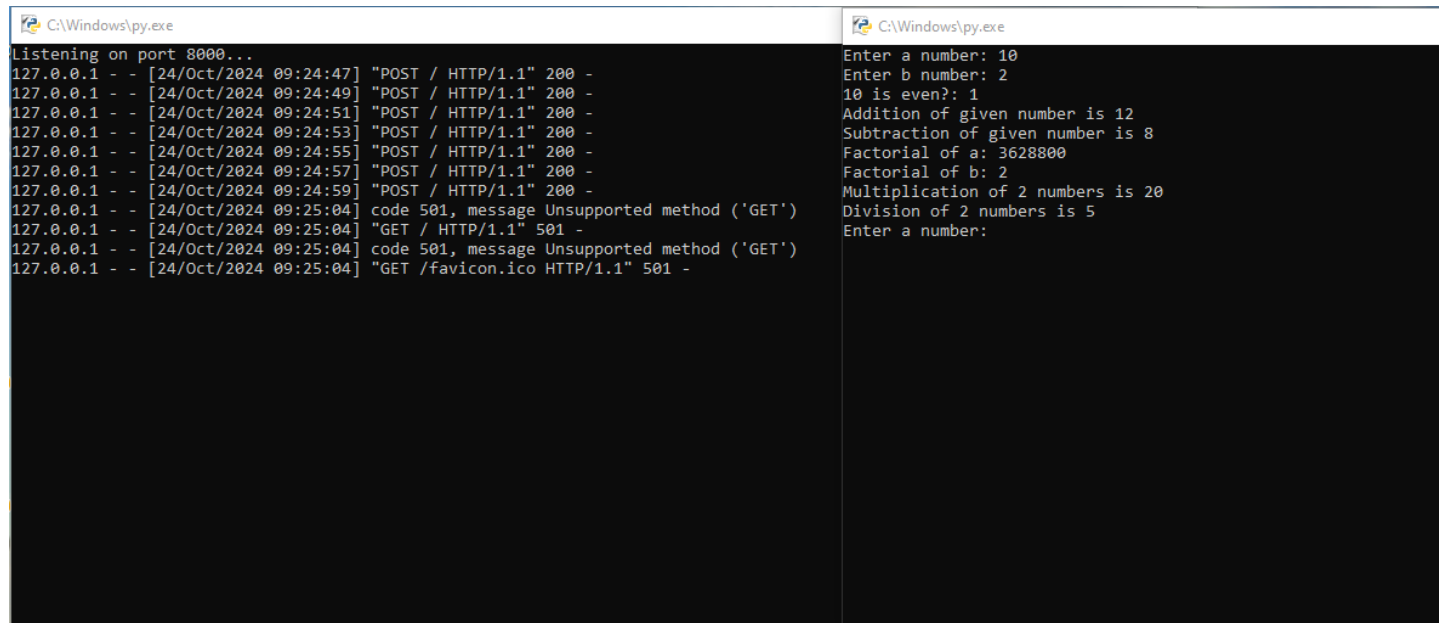
```
from xmlrpc.server import SimpleXMLRPCServer
# Define a function
def is_even(n):
    return n % 2 == 0
def add(a,b):
    return a+b
def sub(a,b):
    return a-b
def factorial(n):
    factorial=1
    for i in range(1,n+1):
        factorial = factorial*i
    return factorial
def multiply(x, y):
    return x * y
def divide(x, y):
    return x // y
# Create server
server = SimpleXMLRPCServer(("localhost", 8000))
print("Listening on port 8000...")
# Register a function under a different name
server.register_function(is_even, "is_even")
server.register_function(add, "add")
server.register_function(sub, "sub")
server.register_function(factorial, "factorial")
#server.register_function(factorial, "factorial")
server.register_function(multiply, 'multiply')
server.register_function(divide, 'divide')
# Run the server's main loop
server.serve_forever()
```

XML RPC PROGRAM- CLIENT SIDE:

```
import xmlrpc.client
proxy= xmlrpc.client.ServerProxy('http://localhost:8000/') # local server
for i in range(5):
    a=int(input("Enter a number:"))
    b=int(input("Enter b number:"))
    print("%d is even?: %d" % (a, (proxy.is_even(a)))) #access XML-RPC server through proxy
```

```
print("addition of given number is %d "%((proxy.add(a,b)))) print("sub of given number is %d\n"%((proxy.sub(a,b)))) print("factorial: %d" %((proxy.factorial(a))))  
print("factorial: %d" %((proxy.factorial(b))))  
print("Multiplication of 2 numbers is %d" %(proxy.multiply(a,b)) print("Division of 2 numbers is %d" %(proxy.divide(a,b))
```

Output



The image shows two side-by-side screenshots of a Windows command prompt window. The left window shows the output of a Python proxy server running on port 8000. It receives several POST requests from 127.0.0.1 and responds with 200 status codes. It also receives GET requests for unsupported methods and responds with 501 status codes. The right window shows the output of a Python script that interacts with the proxy server. It prompts the user to enter a number (10) and a b number (2). It then prints the results of the proxy's add, sub, factorial, multiply, and divide operations. The output shows that the proxy correctly calculated the addition (12), subtraction (8), factorial of 10 (3628800), multiplication (20), and division (5).

```
C:\Windows\py.exe  
Listening on port 8000...  
127.0.0.1 - - [24/Oct/2024 09:24:47] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:24:49] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:24:51] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:24:53] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:24:55] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:24:57] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:24:59] "POST / HTTP/1.1" 200 -  
127.0.0.1 - - [24/Oct/2024 09:25:04] code 501, message Unsupported method ('GET')  
127.0.0.1 - - [24/Oct/2024 09:25:04] "GET / HTTP/1.1" 501 -  
127.0.0.1 - - [24/Oct/2024 09:25:04] code 501, message Unsupported method ('GET')  
127.0.0.1 - - [24/Oct/2024 09:25:04] "GET /favicon.ico HTTP/1.1" 501 -  
  
C:\Windows\py.exe  
Enter a number: 10  
Enter b number: 2  
10 is even?: 1  
Addition of given number is 12  
Subtraction of given number is 8  
Factorial of a: 3628800  
Factorial of b: 2  
Multiplication of 2 numbers is 20  
Division of 2 numbers is 5  
Enter a number:
```