

### REMOTE PROCEDURE CALL FOR LIST OPERATIONS- XMLRPC

**Aim:**

To Implement an XML RPC code for the following functions, a. No of items in a list

**Algorithm:**

**Server side:**

1. Import `SimpleXMLRPCServer`.
2. Define list functions (`length`, `maximum`, `minimum`, `to\_set`, `concatenate`).
3. Create server on `localhost` with port `8000`.
4. Print "Listening on port 8000...".
5. Register functions with the server.
6. Start the server with `serve\_forever()`.
7. Server listens and responds to client requests.

**Client side:**

1. Import ``xmlrpc.client`` to interact with the XML-RPC server.
2. Create a ``ServerProxy`` object to connect to the XML-RPC server at ``http://localhost:8000/``.
3. Enter a loop to repeatedly prompt the user for input to start or stop operations.
4. If the user chooses to start (option 1), prompt the user to input elements for two separate lists (``a`` and ``b``), stopping when ``-1`` or ``-2`` is entered.
5. If the user chooses to stop (option 2), exit the loop.
6. Once the lists are gathered, print the contents of both lists (``a`` and ``b``).
7. Call the registered XML-RPC functions (``list_length``, ``list_maximum``, ``list_minimum``, ``list_to_set``, ``list_concate``) via the server proxy and print the results.

## Program

### Server Side:

```
from xmlrpc.server import SimpleXMLRPCServer
def list_length(a):
    return len(a)
def list_maximum(a):
    return max(a)
def list_minimum(a):
    return min(a)
def list_to_set(a):
    f=list(set(a))
    return f
def list_concate(a,b):
    return a+b
server = SimpleXMLRPCServer(("localhost", 8000)) print("Listening on port
8000...") server.register_function(list_length,"list_length")
server.register_function(list_maximum, "list_maximum")
server.register_function(list_minimum, "list_minimum")
server.register_function(list_to_set, "list_to_set")
server.register_function(list_concate, "list_concate")
server.serve_forever()
```

### Client Side:

```
import xmlrpc.client
proxy= xmlrpc.client.ServerProxy('http://localhost:8000/') while True:
    print("PRESS 1-->STRAT || 2--> STOP ")
    c=int(input("ENTER YOUR CHOICE"))
    a=[]
    b=[]
    if c==1:
        print("ENTER THE ELEMENTS TO ADD FIRST LIST") print("PRESS -1 TO
EXIT THIS LIST")
        while True:
            d=int(input("--->"))
            if d== -1:
                break
            a.append(d)
        print("ENTER THE ELEMENTS TO ADD SECOND LIST") print("PRESS -2 TO
EXIT THIS LIST")
        while True:
            e=int(input("--->"))
            if e== -2:
                break
            b.append(e)
        if c==2:
            break
    print(a)
    print(b)
    print("list_length",proxy.list_length(a))
    print("list_maximum",proxy.list_maximum(a))
    print("list_minimum",proxy.list_minimum(a))
    print("list_to_set",proxy.list_to_set(a))
    print("list_concate",proxy.list_concate(a,b))
```

### Output:

The screenshot shows the Visual Studio Code editor with the following content:

**Project Explorer (Left):**

- src
  - selectionSortClient.py (Selected)

**Editor (Center):**

```

1 import socket
2 proxy = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
3 while True:
4     print("PRESS 1->START || 2-> STOP ")
5     exit(input("ENTER YOUR CHOICE"))
6     a=[]
7     b=[]
8     i=0
9     print("ENTER THE ELEMENTS TO ADD FIRST LIST")
10    print("PRESS -1 TO EXIT THIS LIST")
11    while True:
12        a.append(input("---->"))
13
14    --->5
15    --->4
16    --->3
17    [10, 20, 30, 40]
18    [1, 2, 3, 4]
19    list_length a
20    list_maximum 40
21    list_minimum 10
22    list_to_net [40, 10, 20, 30]
23    list_concat [10, 20, 30, 40, 1, 2, 3, 4]
24    PRESS 1->START || 2-> STOP
25    ENTER YOUR CHOICE2
26
27    Process finished with exit code 0
  
```

**Bottom Status Bar:** Process finished with exit code 0

The screenshot shows a VS Code editor with a Python script named `Client.py` in the `Scripts` directory of a virtual environment. The script uses the `xmlrpc.client` module to connect to a server at `http://localhost:8000/`. The client prompts the user to enter choices and lists of elements. The output window shows the execution of the script, displaying the prompts and user inputs.

```

1  import xmlrpc.client
2  proxy = xmlrpc.client.ServerProxy('http://localhost:8000/')
3  while True:
4      print("PRESS 1-->STRAT || 2--> STOP")
5      exit(input("ENTER YOUR CHOICE"))
6      a=[]
7      b=[]
8      if a==[]:
9          print("ENTER THE ELEMENTS TO ADD FIRST LIST")
10         print("PRESS -1 TO EXIT THIS LIST")
11         while True:
12             a.append(input("---->"))

```

The output window shows the execution of the script, displaying the prompts and user inputs:

```

D:\List XML\venv\Scripts\python.exe "D:\List XML\Client.py"
PRESS 1-->STRAT || 2--> STOP
ENTER YOUR CHOICE:1
ENTER THE ELEMENTS TO ADD FIRST LIST
PRESS -1 TO EXIT THIS LIST
--->10
--->20
--->30
--->40
--->5
ENTER THE ELEMENTS TO ADD SECOND LIST
PRESS -2 TO EXIT THIS LIST
--->1
--->2
--->3

```

The screenshot displays a Python IDE with a project named 'list XML'. The file explorer on the left shows the project structure, including a 'venv' directory and a 'Server.py' file. The main editor window shows the code for 'Server.py', which implements a SimpleXMLRPCServer. The code defines several functions: list\_length, list\_maximum, list\_minimum, list\_to\_set, and list\_concat. The server is initialized to listen on port 8000 and registers these functions. The output window at the bottom shows the server's execution, indicating it is listening on port 8000 and receiving multiple POST requests from 127.0.0.1.

```
1 from xmlrpc.server import SimpleXMLRPCServer
2
3 def list_length(a):
4     return len(a)
5
6 def list_maximum(a):
7     return max(a)
8
9 def list_minimum(a):
10    return min(a)
11
12 def list_to_set(a):
13    return set(a)
14
15 def list_concat(a,b):
16    return a+b
17
18 server = SimpleXMLRPCServer(("localhost", 8000))
19 print("Listening on port 8000...")
20 server.register_function(list_length, name="list_length")
21 server.register_function(list_maximum, name="list_maximum")
22 server.register_function(list_minimum, name="list_minimum")
23
24 server.register_function(list_to_set, name="list_to_set")
25 server.register_function(list_concat, name="list_concat")
```

```
"D:\\list XML\\venv\\Scripts\\python.exe" "D:\\list XML\\Server.py"
Listening on port 8000...
127.0.0.1 - - [07/Nov/2024 08:38:12] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [07/Nov/2024 08:38:14] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [07/Nov/2024 08:38:17] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [07/Nov/2024 08:38:19] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [07/Nov/2024 08:38:21] "POST / HTTP/1.1" 200 -
```

## Result:

Thus, the list operations using Remote Procedure Call was executed.