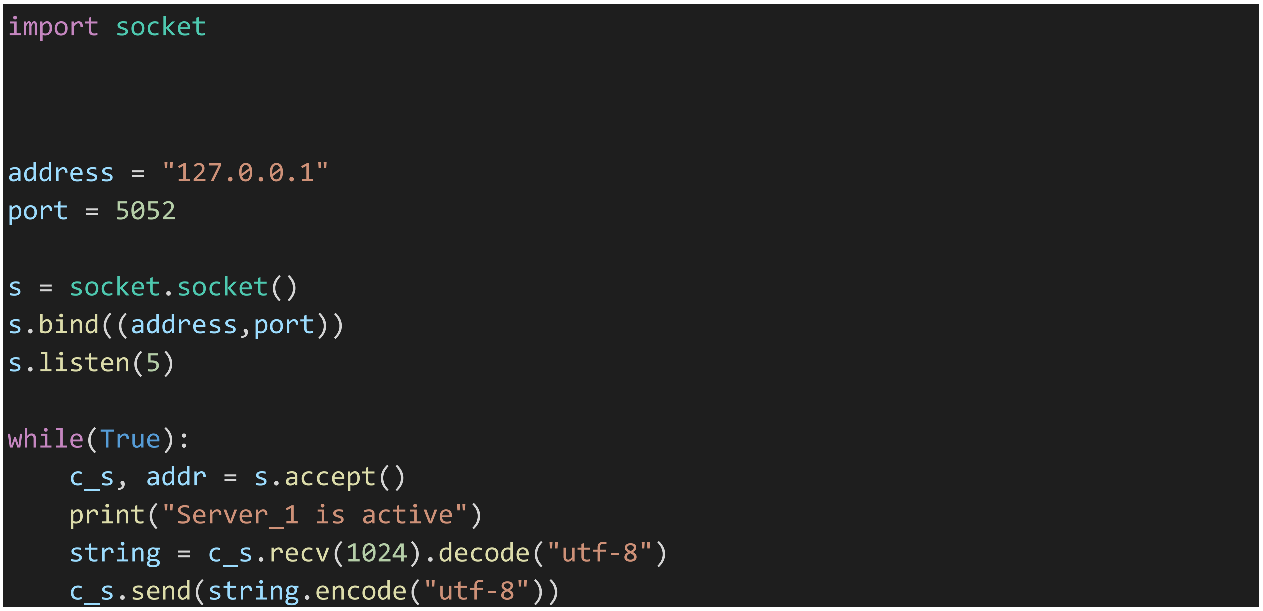
Abdullah
1802016

Main Server (Code)

```
import socket
from Thread import Mythread
import json

address = "127.0.0.1"
port = 5051

s = socket.socket()
s.bind((address,port))
s.listen(5)






while(True):
    print("Main server is Active")
    c_s, addr = s.accept()
    client_thread = Mythread(c_s)
    client_thread.start()






c_s.close()
```

Abdullah
1802016

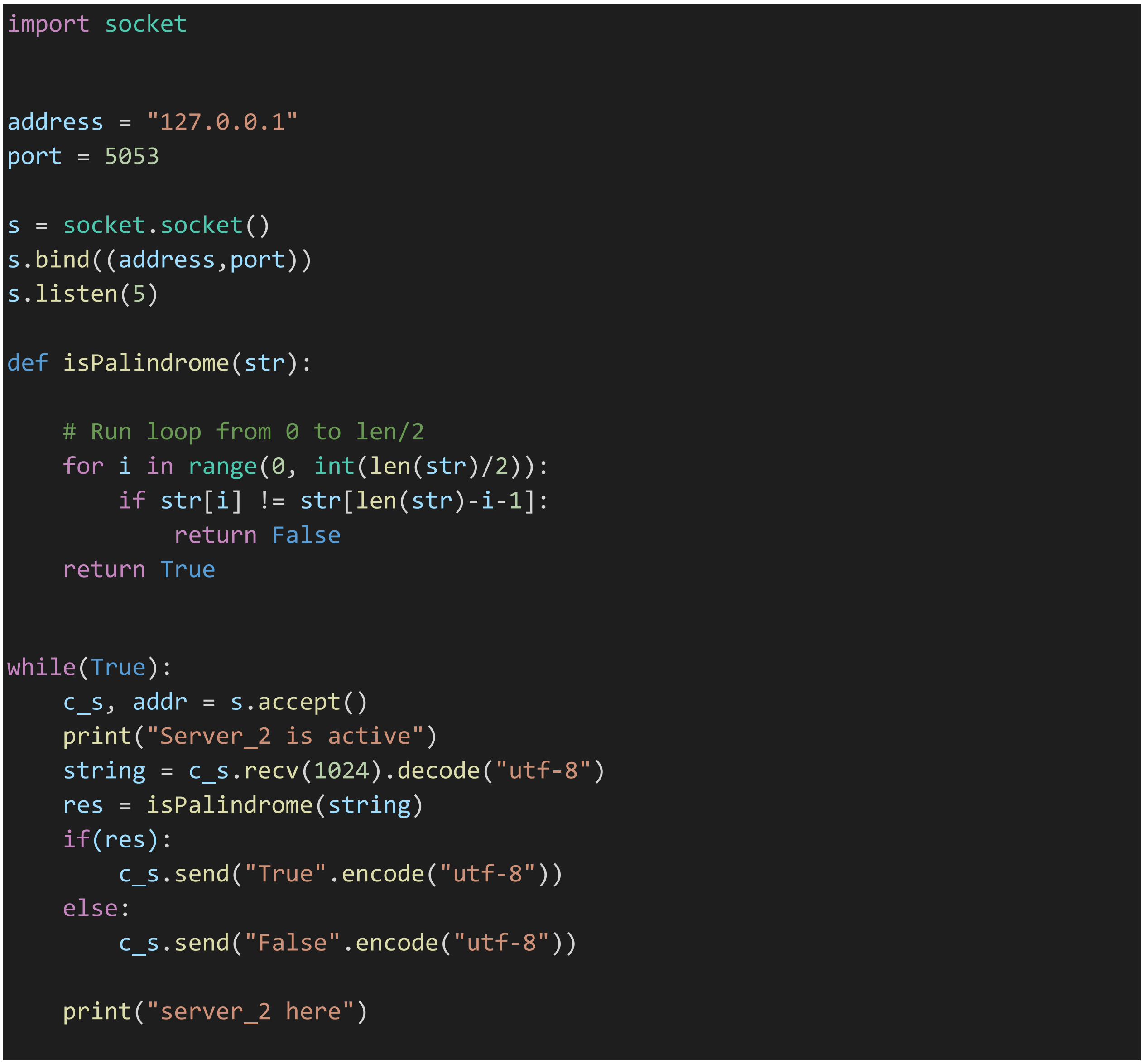
Server 1



```
import socket



address = "127.0.0.1"
port = 5052

s = socket.socket()
s.bind((address,port))
s.listen(5)

while(True):
    c_s, addr = s.accept()
    print("Server_1 is active")
    string = c_s.recv(1024).decode("utf-8")
    c_s.send(string.encode("utf-8"))
```

Abdullah
1802016

Server 2



```python
import socket


address = "127.0.0.1"
port = 5053

s = socket.socket()
s.bind((address,port))
s.listen(5)

def isPalindrome(str):

    # Run loop from 0 to len/2
    for i in range(0, int(len(str)/2)):
        if str[i] != str[len(str)-i-1]:
            return False
    return True


while(True):
    c_s, addr = s.accept()
    print("Server_2 is active")
    string = c_s.recv(1024).decode("utf-8")
    res = isPalindrome(string)
    if(res):
        c_s.send("True".encode("utf-8"))
    else:
        c_s.send("False".encode("utf-8"))

    print("server_2 here")
```

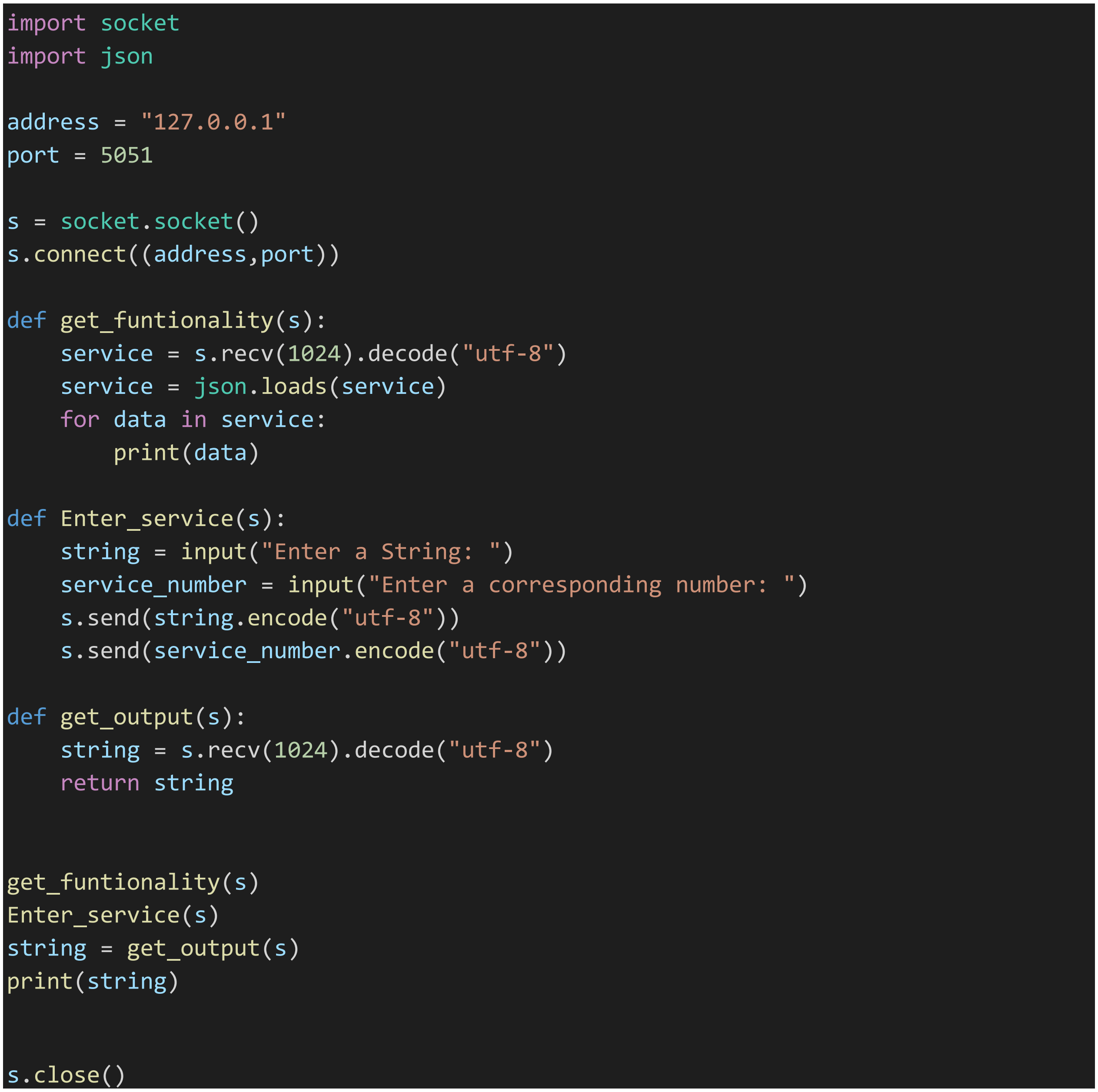
Abdullah
1802016

Server 3

```
import socket


address = "127.0.0.1"
port = 5054

s = socket.socket()
s.bind((address,port))
s.listen(5)

while(True):
    c_s, addr = s.accept()
    print("Server_3 is active")
    string = c_s.recv(1024).decode("utf-8")
    length = str(len(string))
    c_s.send(length.encode("utf-8"))
    print("server_3 here")
```

Abdullah
1802016

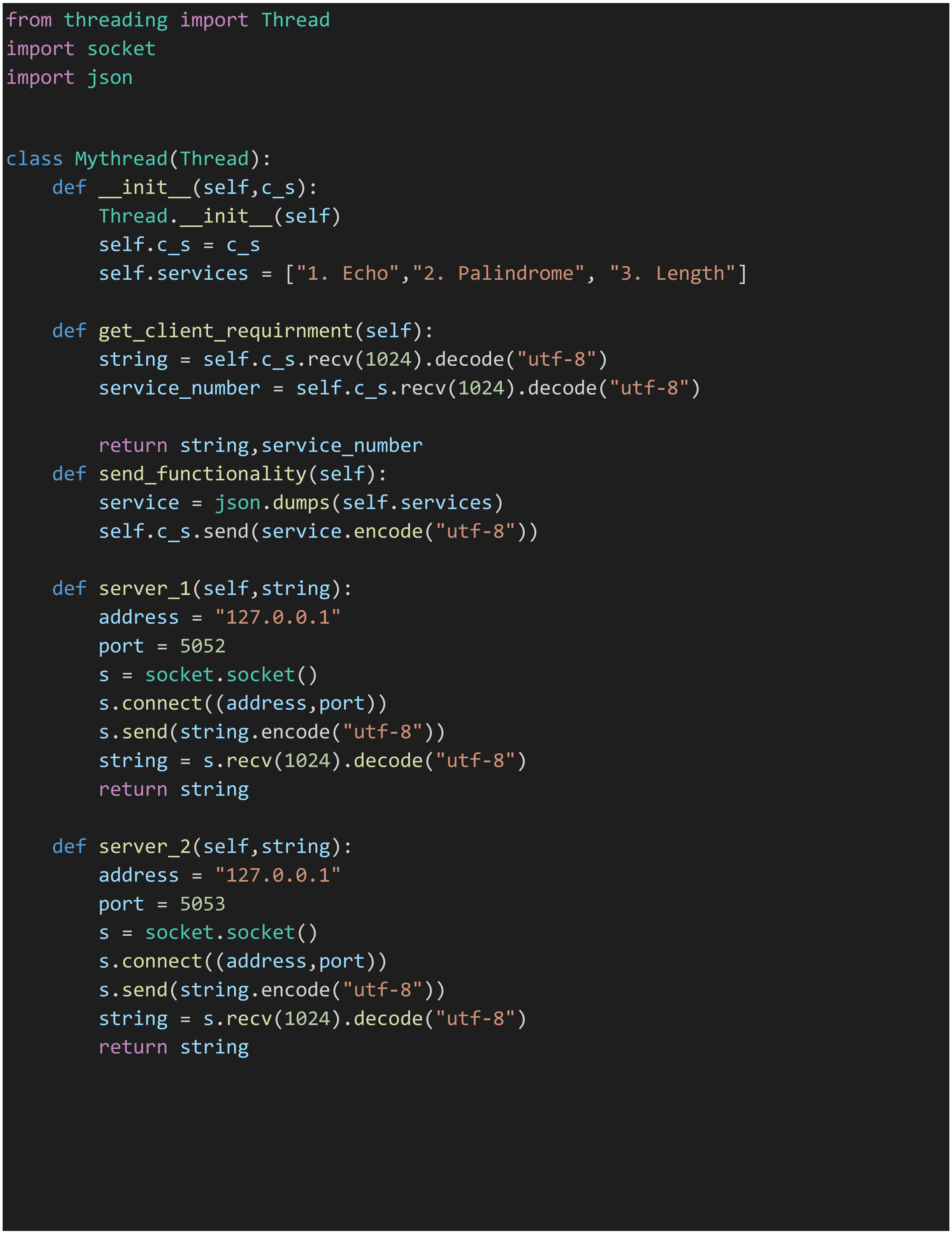
Client



```python
import socket
import json

address = "127.0.0.1"
port = 5051

s = socket.socket()
s.connect((address,port))

def get_funtionality(s):
    service = s.recv(1024).decode("utf-8")
    service = json.loads(service)
    for data in service:
        print(data)

def Enter_service(s):
    string = input("Enter a String: ")
    service_number = input("Enter a corresponding number: ")
    s.send(string.encode("utf-8"))
    s.send(service_number.encode("utf-8"))

def get_output(s):
    string = s.recv(1024).decode("utf-8")
    return string


get_funtionality(s)
Enter_service(s)
string = get_output(s)
print(string)


s.close()
```

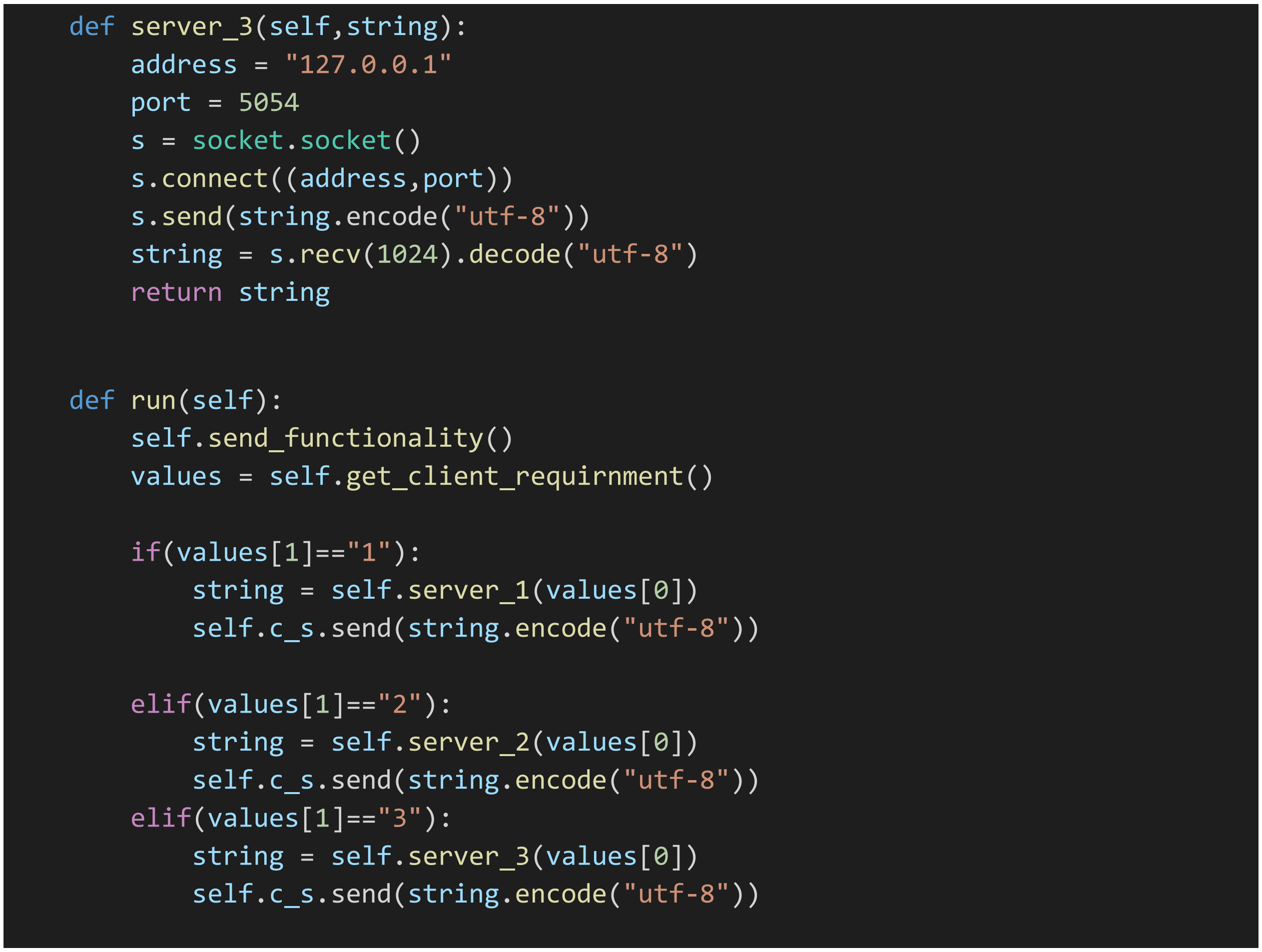
Abdullah
1802016

Thread



```
from threading import Thread
import socket
import json


class Mythread(Thread):
    def __init__(self,c_s):
        Thread.__init__(self)
        self.c_s = c_s
        self.services = ["1. Echo","2. Palindrome", "3. Length"]

    def get_client_requirnment(self):
        string = self.c_s.recv(1024).decode("utf-8")
        service_number = self.c_s.recv(1024).decode("utf-8")

        return string,service_number
    def send_functionality(self):
        service = json.dumps(self.services)
        self.c_s.send(service.encode("utf-8"))

    def server_1(self,string):
        address = "127.0.0.1"
        port = 5052
        s = socket.socket()
        s.connect((address,port))
        s.send(string.encode("utf-8"))
        string = s.recv(1024).decode("utf-8")
        return string

    def server_2(self,string):
        address = "127.0.0.1"
        port = 5053
        s = socket.socket()
        s.connect((address,port))
        s.send(string.encode("utf-8"))
        string = s.recv(1024).decode("utf-8")
        return string
```

Abdullah
1802016



```
    def server_3(self,string):
        address = "127.0.0.1"
        port = 5054
        s = socket.socket()
        s.connect((address,port))
        s.send(string.encode("utf-8"))
        string = s.recv(1024).decode("utf-8")
        return string


    def run(self):
        self.send_functionality()
        values = self.get_client_requirnment()

        if(values[1]=="1"):
            string = self.server_1(values[0])
            self.c_s.send(string.encode("utf-8"))

        elif(values[1]=="2"):
            string = self.server_2(values[0])
            self.c_s.send(string.encode("utf-8"))
        elif(values[1]=="3"):
            string = self.server_3(values[0])
            self.c_s.send(string.encode("utf-8"))
```