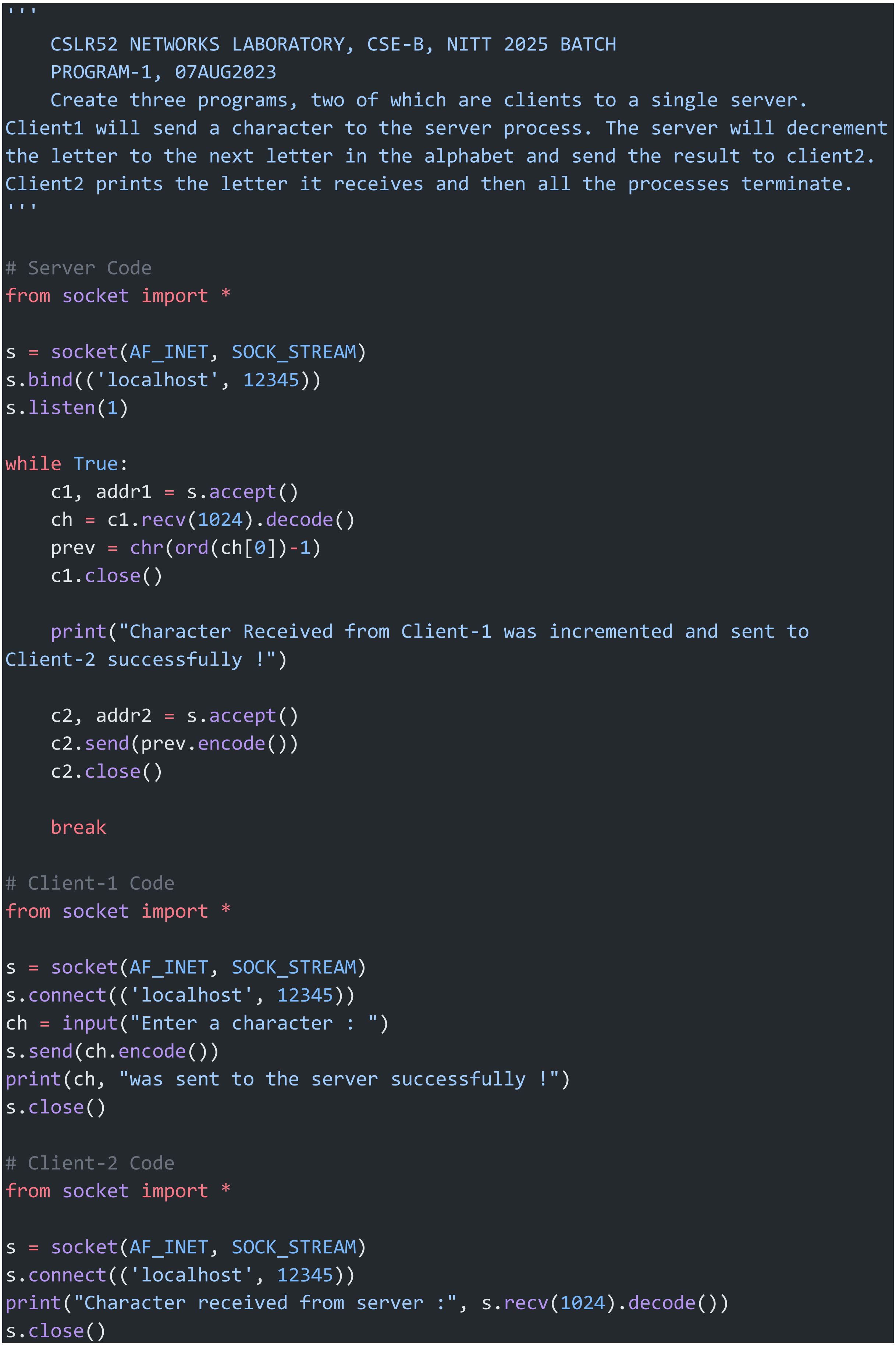
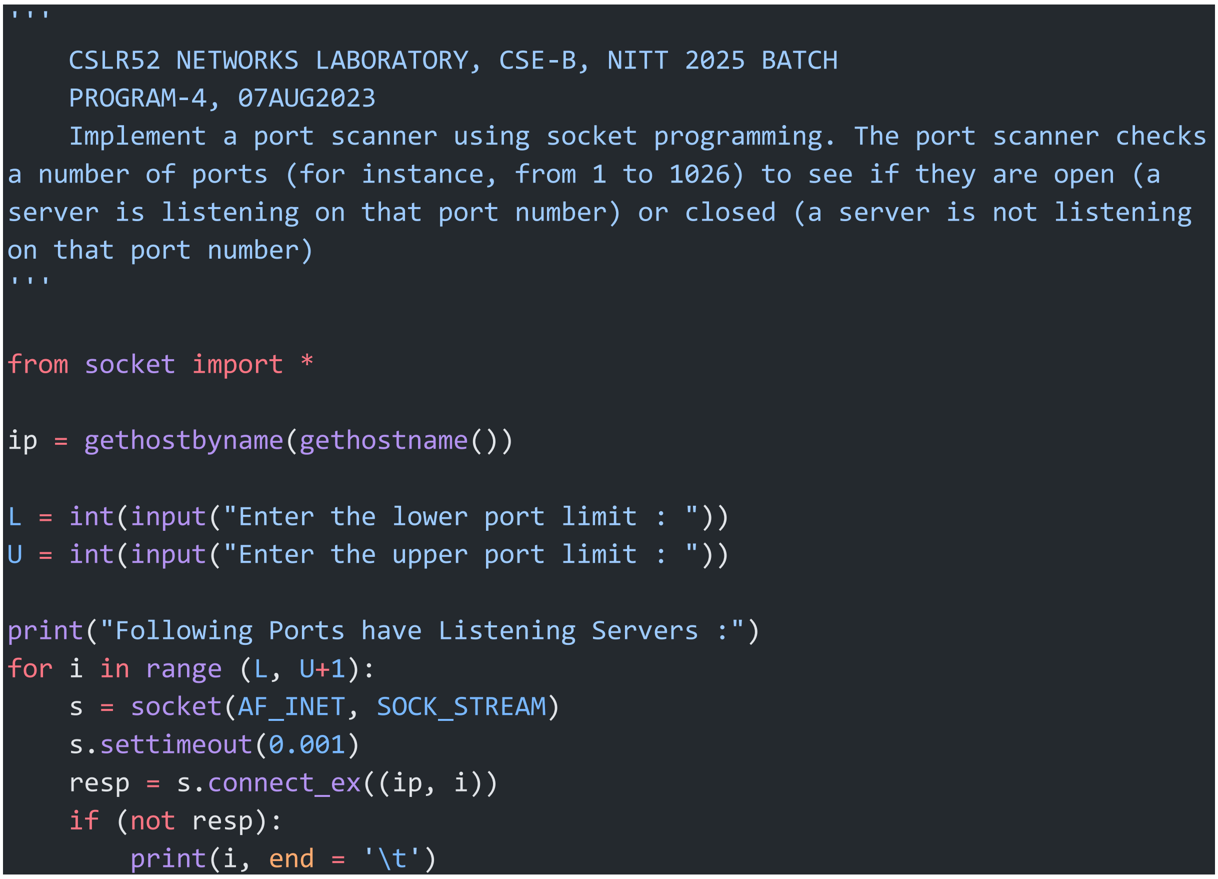


```
'''
    CSLR52 NETWORKS LABORATORY, CSE-B, NITT 2025 BATCH
    PROGRAM-1, 07AUG2023
    Create three programs, two of which are clients to a single server. Client1 will send a character to the server process. The server will decrement the letter to the next letter in the alphabet and send the result to client2. Client2 prints the letter it receives and then all the processes terminate.
'''

# Server Code
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.bind(('localhost', 12345))
s.listen(1)

while True:
    c1, addr1 = s.accept()
    ch = c1.recv(1024).decode()
    prev = chr(ord(ch[0])-1)
    c1.close()

    print("Character Received from Client-1 was incremented and sent to Client-2 successfully !")

    c2, addr2 = s.accept()
    c2.send(prev.encode())
    c2.close()

    break

# Client-1 Code
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.connect(('localhost', 12345))
ch = input("Enter a character : ")
s.send(ch.encode())
print(ch, "was sent to the server successfully !")
s.close()

# Client-2 Code
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.connect(('localhost', 12345))
print("Character received from server :", s.recv(1024).decode())
s.close()
```

```
'''
    CSLR52 NETWORKS LABORATORY, CSE-B, NITT 2025 BATCH
    PROGRAM-2, 07AUG2023
    Write a socket program to enable client1 to send a float value to the
server. The server process should increase the value of the number it receives
by a power of 1.5. The server should print both the value it receives and the
value that it sends. Client2 should print the value it receives from the
server.
'''
# Server Code
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.bind(('localhost', 12345))
s.listen(1)

while True:
    c1, addr1 = s.accept()
    n = float(c1.recv(1024).decode())
    N = n ** 1.5
    c1.close()

    print("Number Received from Client-1 (" + str(n) + ") was modified and
sent to Client-2 (" + str(N) + ") successfully !")

    c2, addr2 = s.accept()
    c2.send(str(N).encode())
    c2.close()

    break

# Client-1 Code
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.connect(('localhost', 12345))
n = input("Enter a floating point number : ")
s.send(n.encode())
print(n, "was sent to the server successfully !")
s.close()

# Client-2 Code
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.connect(('localhost', 12345))
print("Number received from server :", s.recv(1024).decode())
s.close()
```

```
'''
    CSLR52 NETWORKS LABORATORY, CSE-B, NITT 2025 BATCH
    PROGRAM-3, 07AUG2023
    Send datagrams with two arrays of integers (only even numbers) to a
server. The server should check the data, whether there are odd and/or
fraction numbers. If it is not, the server sums the elements of each array and
puts the sum in a third array that is returned to the client. If the server
discovers that the arrays have erroneous then the server does not reply. A
timeout period should be established by the client such that retransmission
occurs after the period expires.
'''

# Server Code
from socket import *

s = socket(AF_INET, SOCK_DGRAM)
s.bind(('localhost', 12345))

count = 0

while True:
    data, addr = s.recvfrom(1024)
    n, sum, sd = 0, 0, 0
    flag = True
    for ch in data.decode():
        if ch == ',':
            if n % 2 != 0:
                flag = False
                break
            else:
                sum += n
                n = 0
        elif ch == '|':
            sd = sum
            sum = 0
        else:
            n = (n*10) + int(ch)

    if flag:
        s.sendto(("Sum of Elements of Array : " + str(sd) + "," +
str(sum)).encode(), addr)
        print("Sums computed and sent successfully !")
        s.close()
        break

    elif count == 5:
        s.sendto("Maximum limit reached. Please send correct input
!".encode(), addr)
```

```
        print("Maximum limit reached. Please send correct input !")
        break

    else:
        count += 1
        s.sendto("".encode(), addr)
        print("Invalid input received.. trying again....")

# Client Code
from socket import *
from time import sleep

s = socket(AF_INET, SOCK_DGRAM)
s.connect(('localhost', 12345))

str1 = input("Enter an array of even integers separated by commas : ")
str2 = input("Enter another such array : ")

while True:
    s.sendto((str1 + "," + "|" + str2 + ",").encode(), ('localhost', 12345))
    data, addr = s.recvfrom(1024)
    msg = data.decode()

    if msg != '':
        print(msg)
        break

    sleep(1)
    print("No response received from server. Retransmitting data....")

s.close()
```



```
'''
    CSLR52 NETWORKS LABORATORY, CSE-B, NITT 2025 BATCH
    PROGRAM-4, 07AUG2023
    Implement a port scanner using socket programming. The port scanner checks
a number of ports (for instance, from 1 to 1026) to see if they are open (a
server is listening on that port number) or closed (a server is not listening
on that port number)
'''

from socket import *

ip = gethostbyname(gethostname())

L = int(input("Enter the lower port limit : "))
U = int(input("Enter the upper port limit : "))

print("Following Ports have Listening Servers :")
for i in range (L, U+1):
    s = socket(AF_INET, SOCK_STREAM)
    s.settimeout(0.001)
    resp = s.connect_ex((ip, i))
    if (not resp):
        print(i, end = '\t')
```