CS359 Computer Networks - Assignment 10

Name: M Maheeth Reddy	Roll No.: 1801CS31	Date: 12-Apr-2021

Question 1:

Introduction: The objective is to write TCP client-server C programs to pass a string from client to server. The client takes a string/message as input and sends it to the server. We need to display the lowercase message at the server side and convert the message from lower case to upper case and display the converted message on the client side.1

I have written two programs, **client1.c** and **server1.c**

Compilation: gcc client1.c –o client1   gcc server1.c –o server1

Execution: ./client1   ./server1

Sample Session 1:

Client Side	Server Side

```
┌─[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:26:28  IST]
└─>$ ./client1
Enter a message: HeLlO, daRknesS my Old FRieND :)
[+]Server socket created successfully.
[+]Connected to Server.

Client sent "HeLlO, daRknesS my Old FRieND :)" to server

Received message in uppercase from server:
      HELLO, DARKNESS MY OLD FRIEND :)

[+]Closing the connection.
```

```
┌─[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:24:36  IST]
└─>$ ./server1
[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....

Received the message "HeLlO, daRknesS my Old FRieND :)" from client
Message in lowercase: hello, darkness my old friend :)
Sending Message in Uppercase to client

[+]Closing the connection.
```

Sample Session 2:

Client Side	Server Side

```
┌─[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:48:55  IST]
└─>$ ./client1
Enter a message: lowercase tO UPPERCASE
[+]Server socket created successfully.
[+]Connected to Server.

Client sent "lowercase tO UPPERCASE" to server

Received message in uppercase from server:
      LOWERCASE TO UPPERCASE

[+]Closing the connection.
```

```
┌─[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:48:55  IST]
└─>$ ./server1
[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....

Received the message "lowercase tO UPPERCASE" from client
Message in lowercase: lowercase to uppercase
Sending Message in Uppercase to client

[+]Closing the connection.
```

Question 2:

Introduction: The objective is to write TCP client-server C programs to pass a string from client to server. The client takes a string/message as input and sends it to the server. We need to display the string on the server side and send the reversed string to the client and display it on the client side.2

I have written two programs, **client2.c** and **server2.c**

Compilation: gcc client2.c –o client2   gcc server2.c –o server2

Execution: ./client2   ./server2

Sample Session 1:

Client Side	Server Side

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:42:28  IST]
└>$ ./client2
Enter a string: Assignment 10
[+]Server socket created successfully.
[+]Connected to Server.

Client sent "Assignment 10" to server

Received reversed string from server:
      01 tnemngissA

[+]Closing the connection.
```

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:42:26  IST]
└>$ ./server2
[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....

Received the message "Assignment 10" from client
Reversed string: 01 tnemngissA

Sending reversed string to server

[+]Closing the connection.
```

Sample Session 2:

Client Side	Server Side

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:42:38  IST]
└>$ ./client2
Enter a string: redivider
[+]Server socket created successfully.
[+]Connected to Server.

Client sent "redivider" to server

Received reversed string from server:
      redivider

[+]Closing the connection.
```

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:42:38  IST]
└>$ ./server2
[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....

Received the message "redivider" from client
Reversed string: redivider

Sending reversed string to server

[+]Closing the connection.
```

Note: **redivider** is a palindrome. So, it appears the same even after reversing

Question 3:

Introduction: The objective is to write TCP client-server C programs to send an integer array from client to server. On the server side, the integer array is displayed.

I have written two programs, **client3.c** and **server3.c**

Compilation: gcc client3.c –o client3   gcc server3.c –o server3

Execution: ./client3   ./server3

Sample Session 1:

Client Side	Server Side

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:46:50  IST]
└>$ ./client3
Enter number of integers: 5
Enter integers: 0 -3 5 7 -4

[+]Server socket created successfully.
[+]Connected to Server.

Client sent number of integers (5) to server
Client sent the array of integers to server

[+]Closing the connection.
```

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:46:47  IST]
└>$ ./server3
[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....

Received number of integers (5) from client
Received array of integers from client:
[0, -3, 5, 7, -4]

[+]Closing the connection.
```

Sample Session 2:

Client Side	Server Side

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:47:12  IST]
└>$ ./client3
Enter number of integers: 7
Enter integers: 5 6 -9 0 1 10 2

[+]Server socket created successfully.
[+]Connected to Server.

Client sent number of integers (7) to server
Client sent the array of integers to server

[+]Closing the connection.
```

```
┌[maheeth@maheeth-PC:~/D/W/C/lab10]─[03:47:12  IST]
└>$ ./server3
[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....

Received number of integers (7) from client
Received array of integers from client:
[5, 6, -9, 0, 1, 10, 2]

[+]Closing the connection.
```

----------x-----------------x---------