**Practical 8**

**Aim:**

Implement any one sorting algorithm using TCP/UDP on Server application and Give Input on Client side and client should sorted output from server and display sorted on input side.

**Description:**

****

Fig. 1. Concurrent Client Server senario

**Program code:**

**ClientTCP.java**

//TCP Client

**import** java**.**io**.\*;**

**import** java**.**net**.\*;**

public class ClientTCP **{**

public static void main**(**String**[]** args**)** **throws** IOException **{**

String serverHostname **=** **new** String **(**"127.0.0.1"**);**

**if** **(**args**.**length **>** 0**)**

serverHostname **=** args**[**0**];**

System**.**out**.**println **(**"Attemping to connect to host " **+**

serverHostname **+** " on port 10008."**);**

Socket echoSocket **=** **null;**

PrintWriter out **=** **null;**

BufferedReader in **=** **null;**

**try** **{**

echoSocket **=** **new** Socket**(**serverHostname**,** 10008**);**

out **=** **new** PrintWriter**(**echoSocket**.**getOutputStream**(),** **true);**

in **=** **new** BufferedReader**(new** InputStreamReader**(**

echoSocket**.**getInputStream**()));**

**}** **catch** **(**UnknownHostException e**)** **{**

System**.**err**.**println**(**"Don't know about host: " **+** serverHostname**);**

System**.**exit**(**1**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"Couldn't get I/O for "

**+** "the connection to: " **+** serverHostname**);**

System**.**exit**(**1**);**

**}**

BufferedReader stdIn **=** **new** BufferedReader**(**

**new** InputStreamReader**(**System**.**in**));**

String userInput**;**

System**.**out**.**println **(**"Enter integers to be sorted with space. (\"Bye.\" to quit)"**);**

**while** **((**userInput **=** stdIn**.**readLine**())** **!=** **null)**

**{**

out**.**println**(**userInput**);**

// end loop

**if** **(**userInput**.**equals**(**"Bye."**))**

**break;**

System**.**out**.**println**(**"echo: " **+** in**.**readLine**());**

**}**

out**.**close**();**

in**.**close**();**

stdIn**.**close**();**

echoSocket**.**close**();**

}

}

**ServerTCP.java**

//TCP Server

**import** java**.**net**.\*;**

**import** java**.**io**.\*;**

**import** java**.**util**.\*;**

public class ServerTCP **extends** Thread

**{**

protected Socket clientSocket**;**

public static void main**(**String**[]** args**)** **throws** IOException

**{**

ServerSocket serverSocket **=** **null;**

**try** **{**

serverSocket **=** **new** ServerSocket**(**10008**);**

System**.**out**.**println **(**"Enroll: 130050131063"**);**

System**.**out**.**println **(**"Connection Socket Created"**);**

**try** **{**

**while** **(true)**

**{**

System**.**out**.**println **(**"Waiting for Connection"**);**

**new** ServerTCP **(**serverSocket**.**accept**());**

**}**

**}**

**catch** **(**IOException e**)**

**{**

System**.**err**.**println**(**"Accept failed."**);**

System**.**exit**(**1**);**

**}**

**}**

**catch** **(**IOException e**)**

**{**

System**.**err**.**println**(**"Could not listen on port: 10008."**);**

System**.**exit**(**1**);**

**}**

**finally**

**{**

**try** **{**

serverSocket**.**close**();**

**}**

**catch** **(**IOException e**)**

**{**

System**.**err**.**println**(**"Could not close port: 10008."**);**

System**.**exit**(**1**);**

**}**

**}**

**}**

private ServerTCP**(**Socket clientSoc**)**

**{**

clientSocket **=** clientSoc**;**

start**();**

**}**

public void run**()**

**{**

System**.**out**.**println **(**"New Communication Thread Started"**);**

**try** **{**

PrintWriter out **=** **new** PrintWriter**(**clientSocket**.**getOutputStream**(),**

**true);**

BufferedReader in **=** **new** BufferedReader**(**

**new** InputStreamReader**(** clientSocket**.**getInputStream**()));**

String inputLine**;**

String**[]** inputTemp**;**

**while** **((**inputLine **=** in**.**readLine**())** **!=** **null)** **{**

System**.**out**.**println **(**"Server: " **+** inputLine**);**

**if** **(**inputLine**.**equals**(**"Bye."**))**

**break;**

inputTemp **=** inputLine**.**split**(**" "**);**

int input**[]** **=** **new** int**[**inputTemp**.**length**];**

**for(**int i **=** 0**;** i**<**input**.**length **;** i**++){**

String temp **=** inputTemp**[**i**];**

input**[**i**]** **=** Integer**.**parseInt**(**temp**);**

**}**

int ans**[]** **=** **new** BubbleSort**(**input**,** input**.**length**).**sort**();**

StringBuffer sbAns **=** **new** StringBuffer**();**

**for(**int i**=**0**;**i**<**ans**.**length**;**i**++)** **{**

sbAns**.**append**(**ans**[**i**]+**" "**);**

**}**

//System.out.println ("Server: " + sbAns);

String replyAns **=** sbAns**.**toString**();**

out**.**println**(**"Sorted array is: "**+**replyAns**);**

**}**

out**.**close**();**

in**.**close**();**

clientSocket**.**close**();**

**}**

**catch** **(**IOException e**)**

**{**

System**.**err**.**println**(**"Problem with Communication Server"**);**

System**.**exit**(**1**);**

**}**

**}**

**}**

**ClientUDP.java**

//UDP Client

**import** java**.**net**.\*;**

**import** java**.**io**.\*;**

public class ClientUDP **{**

public static void main**(**String args**[])** **{**

DatagramSocket datagramSocket **=** **null;**

**try** **{**

datagramSocket **=** **new** DatagramSocket**();**

BufferedReader stdIn **=** **new** BufferedReader**(**

**new** InputStreamReader**(**System**.**in**));**

String userInput**;**

System**.**out**.**print **(**"input: "**);**

**while** **((**userInput **=** stdIn**.**readLine**())** **!=** **null)** **{**

**if** **(**userInput**.**equals**(**"Bye."**))**

**break;**

byte**[]** b **=** userInput**.**getBytes**();**

InetAddress host **=** InetAddress**.**getByName**(**"localhost"**);**

DatagramPacket dataPacket **=** **new** DatagramPacket**(**b**,** b**.**length**,** host**,** 10007**);**

// System.out.println ("Attemping to connect to host " +

// dataPacket.getAddress() + " on port 10007.");

datagramSocket**.**send**(**dataPacket**);**

System**.**out**.**println**(**"Client: "**+**userInput**);**

byte**[]** buffer **=** **new** byte**[**50**];**

DatagramPacket reply **=** **new** DatagramPacket**(**buffer**,** buffer**.**length**);**

datagramSocket**.**receive**(**reply**);**

String word **=** **new** String**(**reply**.**getData**());**

System**.**out**.**println**(**"Server: "**+**word**);**

System**.**out**.**print **(**"input: "**);**

**}**

**}** **catch** **(**UnknownHostException e**)** **{**

System**.**err**.**println**(**"Don't know about host: localhost"**);**

System**.**exit**(**1**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"Couldn't get I/O for "

**+** "the connection to: localhost"**);**

e**.**printStackTrace**();**

System**.**exit**(**1**);**

**}**

datagramSocket**.**close**();**

**}** **}**

**ServerUDP.java**

//UDP Server

**import** java**.**net**.\*;**

**import** java**.**io**.\*;**

**import** java**.**util**.\*;**

public class ServerUDP **{**

public static void main**(**String args**[])** **{**

DatagramSocket datagramSocket **=** **null;**

**try** **{**

datagramSocket **=** **new** DatagramSocket**(**10007**);**

System**.**out**.**println **(**"Enroll: 130050131063"**);**

System**.**out**.**println **(**"Waiting for connection....."**);**

String inputLine**;**

String**[]** inputTemp**;**

**while(true)** **{**

byte **[]** buffer **=** **new** byte**[**50**];**

DatagramPacket request **=** **new** DatagramPacket**(**buffer**,** buffer**.**length**);**

datagramSocket**.**receive**(**request**);**

System**.**out**.**println **(**"Connection successful"**);**

System**.**out**.**println **(**"Waiting for input....."**);**

String arrayMsg **=** **new** String**(**request**.**getData**());**

System**.**out**.**println**(**"From: " **+** request**.**getAddress**()** **+** " Port: " **+** request**.**getPort**());**

System**.**out**.**println**(**"Client: "**+**arrayMsg**);**

inputTemp **=** arrayMsg**.**split**(**" "**);**

int input**[]** **=** **new** int**[**inputTemp**.**length**];**

**try** **{**

**for(**int i **=** 0**;** i**<**input**.**length **;** i**++){**

String temp **=** inputTemp**[**i**].**trim**();**

input**[**i**]** **=** Integer**.**parseInt**(**temp**,** 10**);**

**}**

**}** **catch** **(**NumberFormatException e**)** **{**

System**.**exit(1);

**}**

int ans**[]** **=** **new** BubbleSort**(**input**,** input**.**length**).**sort**();**

StringBuffer sbAns **=** **new** StringBuffer**();**

**for(**int i**=**0**;**i**<**ans**.**length**;**i**++)** **{**

sbAns**.**append**(**ans**[**i**]+**" "**);**

**}**

//System.out.println("Server: " + sbAns);

String replyAns **=** sbAns**.**toString**();**

String Msg **=** **new** String**(**"Sorted array is: "**+**replyAns**);**

byte**[]** sendMsg **=** Msg**.**getBytes**();**

DatagramPacket reply **=** **new** DatagramPacket**(**sendMsg**,** sendMsg**.**length**,** request**.**getAddress**(),** request**.**getPort**());**

datagramSocket**.**send**(**reply**);**

String word **=** **new** String**(**reply**.**getData**());**

//System.out.println("Server: "+word);

System**.**out**.**println **(**"Waiting for connection....."**);**

**}**

**}catch(**Exception ex**)** **{**

**}**

datagramSocket**.**close**();**

**}**

**}**

**BubbleSort.java**

//Bubble Sort for array of int

public class BubbleSort **{**

int**[]** array**;**

int size**;**

public BubbleSort**(**int**[]** array**,** int size**)** **{**

**this.**array **=** array**;**

**this.**size **=** size**;**

**}**

public int**[]** sort**()** **{**

**for** **(**int i **=** 0**;** i **<** **(** size **-** 1 **);** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** size **-** i **-** 1**;** j**++)** **{**

**if** **(**array**[**j**]** **>** array**[**j**+**1**])**

**{**

int temp **=** array**[**j**];**

array**[**j**]** **=** array**[**j**+**1**];**

array**[**j**+**1**]** **=** temp**;**

**}**

**}**

**}**

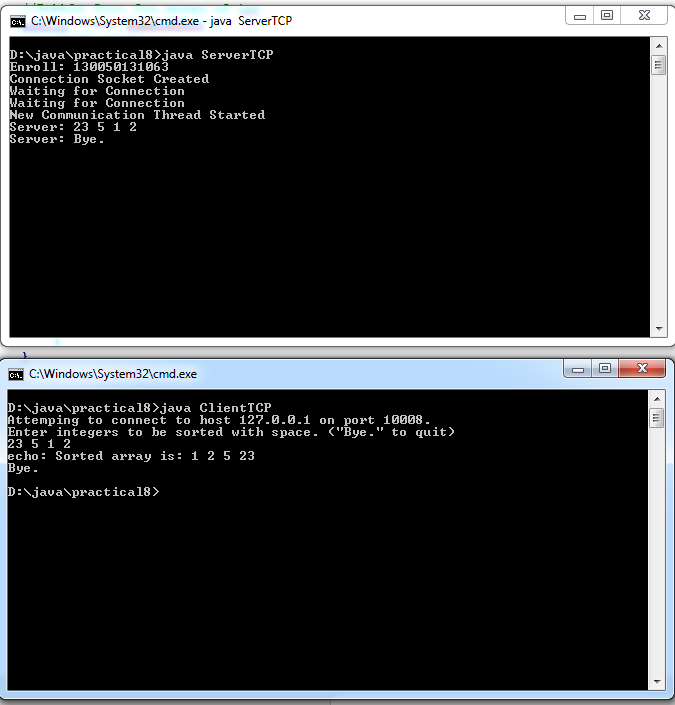
**return** array**;**

**}**

**}**

**Input Output:**

For TCP



For UDP

