Saquib Naushad
1BM19CS144

CN_LAB REPORT

1. Write a program for error detecting code using CRC-CCITT (16-bits).

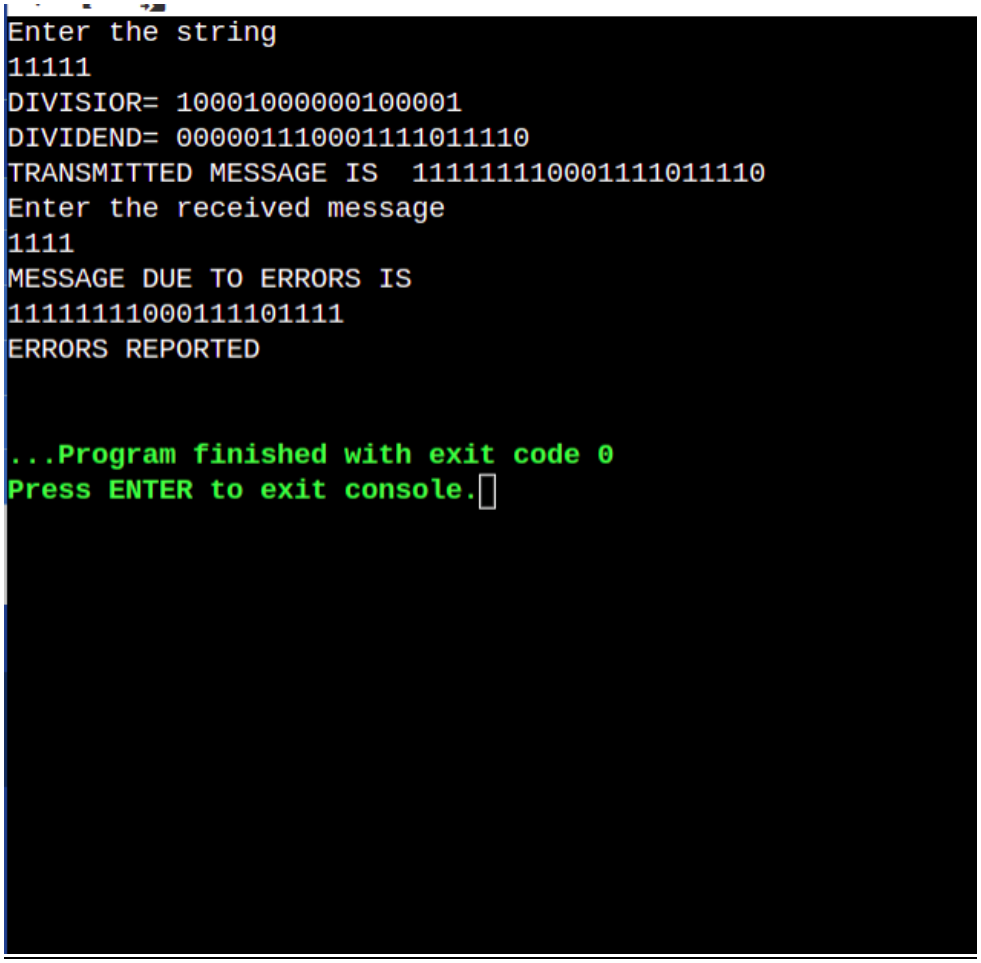
CODE:-

```
Import
java.io.*;
                import java.lang.*;
                import java.util.*;
                class Main
                {
                  public static String string_val(String sts,int poly_length)
                  {
                     for(int i=1;i<poly_length;i++)
                     {
                        sts=sts+"0";
                     }
                     return sts;

                  }

                  public static String generate(char[] divisior,char[] dividend,int len,String org)
                  {
                    for(int i=0;i<len;i++)
                    {
                     if(dividend[i]=='1')
                     {
                       for(int j=0;j<divisior.length;j++)
                       {
                         if(dividend[i+j]==divisior[j])
                         {
                           dividend[i+j]='0';
                         }
                         else
                         {
                          dividend[i+j]='1';
                         }
                       }
                     }
                    }
                     String st=String.valueOf(dividend);

                     String fin=org+st.substring(len);
```

```
    return fin;

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

  String str,rec;
  String d="10001000000100001";
  Scanner sc=new Scanner(System.in);
  System.out.println("Enter the string");
  str=sc.next();

  String org=str;

 int len=str.length();

  str=string_val(str,d.length());

  char[] divisior=d.toCharArray();
  char[] dividend=str.toCharArray();


  String fin=generate(divisior,dividend,len,org);

  System.out.println("DIVISIOR= " + String.valueOf(divisior));
  System.out.println("DIVIDEND= " + String.valueOf(dividend));


   System.out.println("TRANSMITTED MESSAGE IS  " + fin);

   System.out.println("Enter the received message");
   rec=sc.next();
   org=rec;

   len=rec.length();

   rec=string_val(rec,d.length());

   dividend=rec.toCharArray();

   String rin=generate(divisior,dividend,len,org);


   System.out.println("MESSAGE DUE TO ERRORS IS");
   System.out.println(rin);
   if(fin.equals(rin))
   {
      System.out.println("NO ERRORS");
   }
   else
   {
```

```
                System.out.println("ERRORS REPORTED");
            }
        }
        }
```

OUTPUT:-



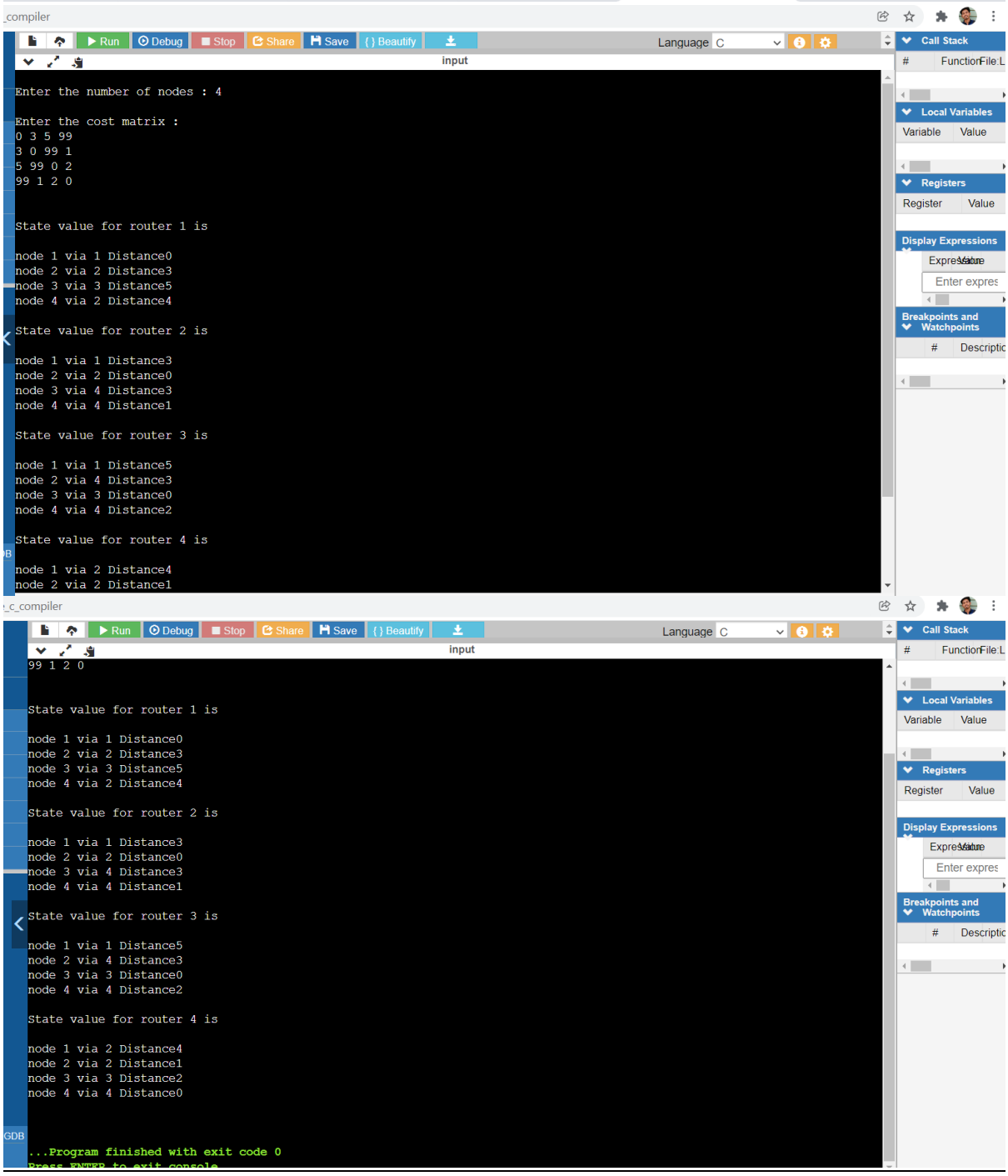
2. Write a program for distance vector algorithm to find suitable path for transmission.

CODE:-

```
#include<stdio.h>
                struct node
                {
                        unsigned dist[20];
                        unsigned from[20];
                }
                rt[10];
                int main()
                {
                        int dmat[20][20];
                        int n,i,j,k,count=0;
                        printf("\nEnter the number of nodes : ");
                        scanf("%d",&n);
                        printf("\nEnter the cost matrix :\n");
                        for(i=0;i<n;i++)
                                for(j=0;j<n;j++)
                                {
                                        scanf("%d",&dmat[i][j]);
                                        dmat[i][i]=0;
                                        rt[i].dist[j]=dmat[i][j];
                                        rt[i].from[j]=j;
                                }
                                do
                                {
                                        count=0;
                                        for(i=0;i<n;i++)
                                        for(j=0;j<n;j++)
```

```
                    for(k=0;k<n;k++)
if(rt[i].dist[j]>dmat[i][k]+rt[k].dist[j])
        {
          rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                rt[i].from[j]=k;
                count++;
        }
                }while(count!=0);
                for(i=0;i<n;i++)
                {
        printf("\n\nState value for router %d is \n",i+1);
                for(j=0;j<n;j++)
                {
printf(" \t\nnode %d via %d Distance%d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
        }
                }
        printf("\n\n");
}
```

OUTPUT:


_compiler
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
input
Enter the number of nodes : 4
Enter the cost matrix :
0 3 5 99
3 0 99 1
5 99 0 2
99 1 2 0
State value for router 1 is
node 1 via 1 Distance0
node 2 via 2 Distance3
node 3 via 3 Distance5
node 4 via 2 Distance4
State value for router 2 is
node 1 via 1 Distance3
node 2 via 2 Distance0
node 3 via 4 Distance3
node 4 via 4 Distance1
State value for router 3 is
node 1 via 1 Distance5
node 2 via 4 Distance3
node 3 via 3 Distance0
node 4 via 4 Distance2
State value for router 4 is
node 1 via 2 Distance4
node 2 via 2 Distance1
Call Stack
Local Variables
Registers
Display Expressions
Breakpoints and Watchpoints
GDB
...Program finished with exit code 0

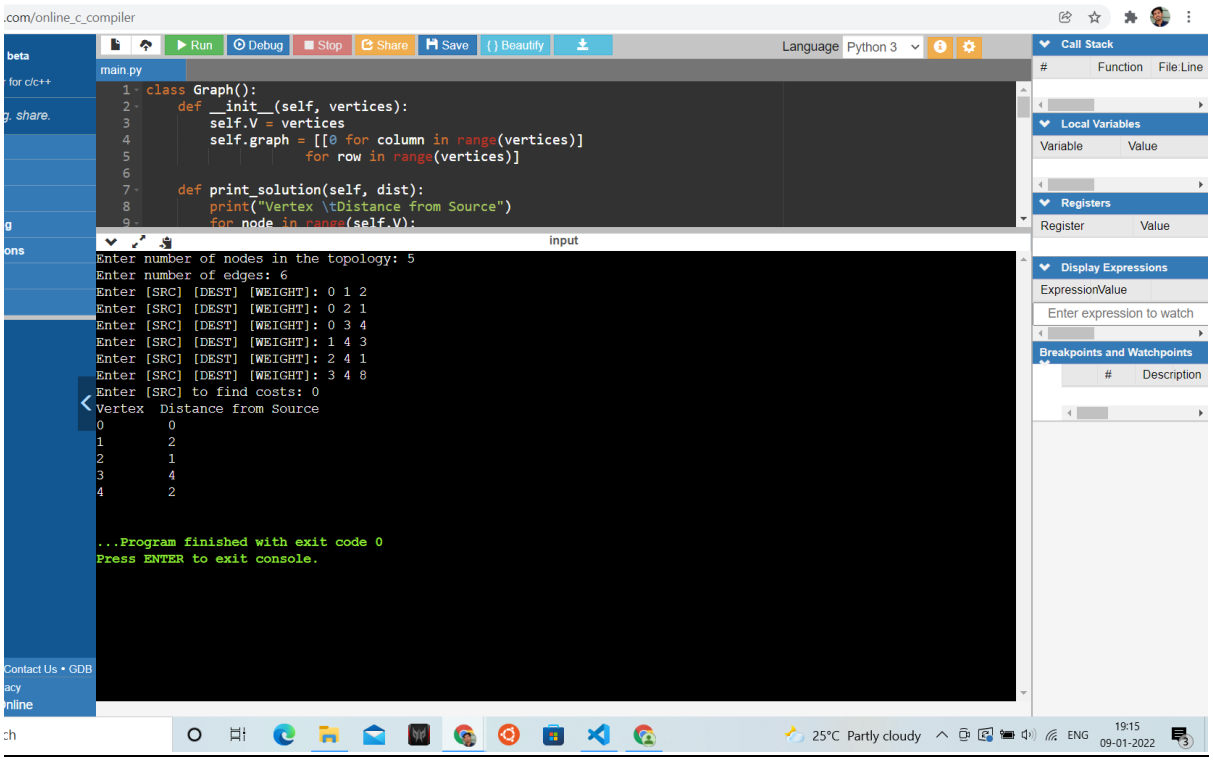
3 . Implement Dijkstra's algorithm to compute the shortest path for a given topology.

CODE:

```
class
Graph():
        def __init__(self, vertices):
            self.V = vertices
            self.graph = [[0 for column in range(vertices)]
                    for row in range(vertices)]

        def print_solution(self, dist):
            print("Vertex \tDistance from Source")
            for node in range(self.V):
                print(node, "\t", dist[node])

        def min_distance(self, dist, sptSet):
            min = 9999
            for v in range(self.V):
                if dist[v] < min and sptSet[v] == False:
                    min = dist[v]
                    min_index = v

            return min_index

        def add_edge(self, src, dest, weight):
            self.graph[src][dest] = self.graph[dest][src] = weight
```

```
    def dijkstra(self, src):

        dist = [9999] * self.V
        dist[src] = 0
        sptSet = [False] * self.V

        for cout in range(self.V):
            u = self.min_distance(dist, sptSet)
            sptSet[u] = True
            for v in range(self.V):
                if self.graph[u][v] > 0 and sptSet[v] == False and dist[v] > dist[u] +
self.graph[u][v]:
                    dist[v] = dist[u] + self.graph[u][v]

        self.print_solution(dist)
g = Graph(int(input("Enter number of nodes in the topology: ")))
e = int(input("Enter number of edges: "))

for i in range(e):
    src, dest, cost = [int(_) for _ in input("Enter [SRC] [DEST] [WEIGHT]: ").split(' ')]
    g.add_edge(src, dest, cost)

src = int(input("Enter [SRC] to find costs: "))
g.dijkstra(src)
```

OUTPUT:


.com/online_c_compiler
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
class Graph():
def __init__(self, vertices):
self.V = vertices
self.graph = [[0 for column in range(vertices)]
for row in range(vertices)]
def print_solution(self, dist):
print("Vertex \tDistance from Source")
for node in range(self.V):
input
Enter number of nodes in the topology: 5
Enter number of edges: 6
Enter [SRC] [DEST] [WEIGHT]: 0 1 2
Enter [SRC] [DEST] [WEIGHT]: 0 2 1
Enter [SRC] [DEST] [WEIGHT]: 0 3 4
Enter [SRC] [DEST] [WEIGHT]: 1 4 3
Enter [SRC] [DEST] [WEIGHT]: 2 4 1
Enter [SRC] [DEST] [WEIGHT]: 3 4 8
Enter [SRC] to find costs: 0
Vertex Distance from Source
0 0
1 2
2 1
3 4
4 2
...Program finished with exit code 0
Press ENTER to exit console.
Call Stack
Function
File:Line
Local Variables
Variable
Value
Registers
Register
Value
Display Expressions
Expression
Value
Enter expression to watch
Breakpoints and Watchpoints
Description
25°C Partly cloudy
ENG
19:15
09-01-2022

4 . Write a program for congestion control using Leaky bucket algorithm.

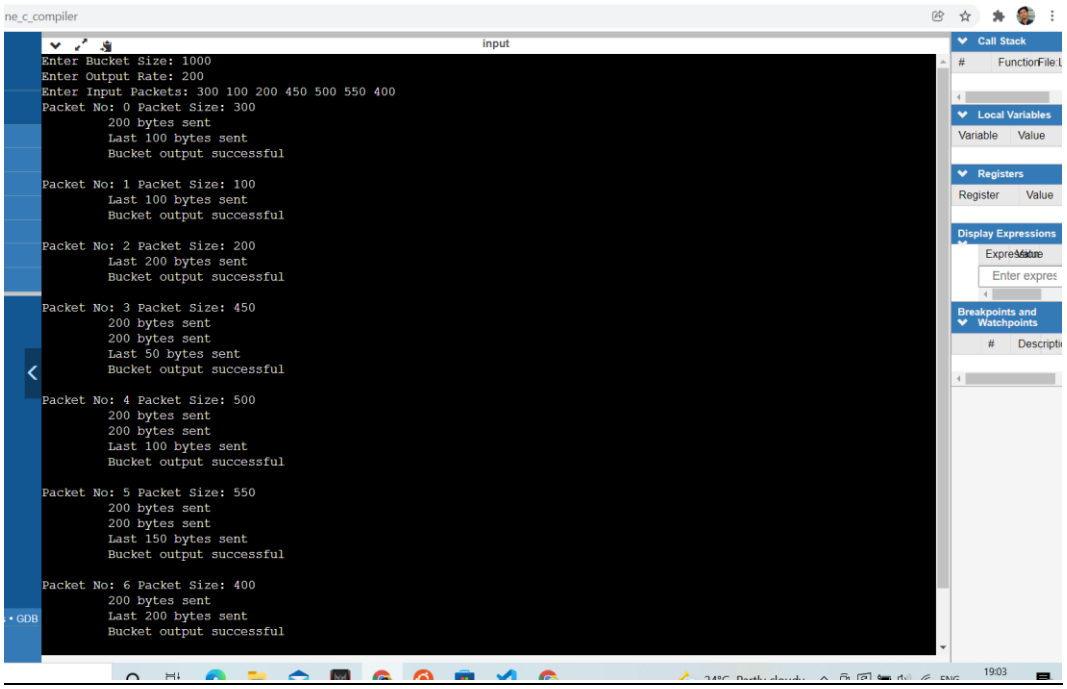
CODE:

```
class
LeakyBucket():
                    def __init__(self, bucket_size, output_rate, input_packets):
                     self.size = bucket_size
                     self.orate = output_rate
                     self.istream = input_packets

                    def congestion_control(self):
                     for x in range(len(self.istream)):
                       packet_size = self.istream[x]
                       print(f"Packet No: {x} Packet Size: {packet_size}")

                       if packet_size > self.size:
                         print("\t Bucket Overflow")
                       else:
                         while packet_size > self.orate:
                           print(f"\t {self.orate} bytes sent")
                           packet_size -= self.orate

                         if packet_size:
                           print(f"\t Last {packet_size} bytes sent")

                         print("\t Bucket output successful \n")

                  bucket_size = int(input("Enter Bucket Size: "))
                  output_rate = int(input("Enter Output Rate: "))
                  input_packets = list(map(int, input("Enter Input Packets: ").split()))
```

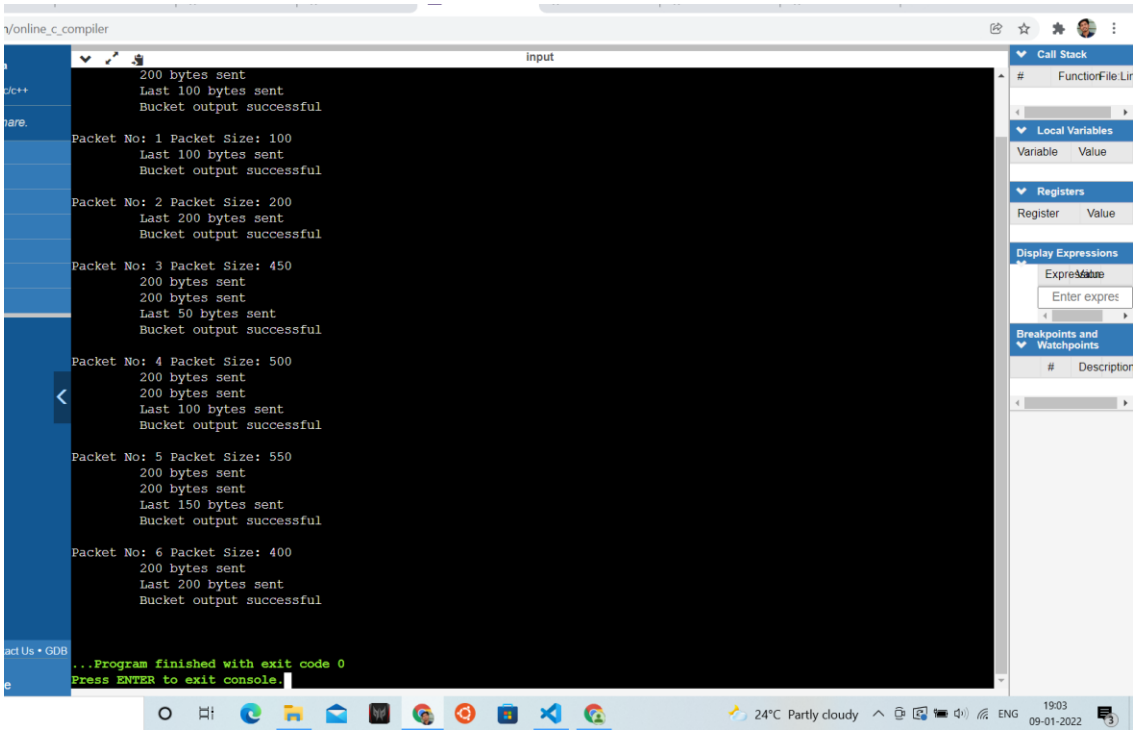
network = LeakyBucket(bucket_size, output_rate, input_packets)

network.congestion_control()

OUTPUT:



5.



5. Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents to the requested file if present.

CODE:

Server:

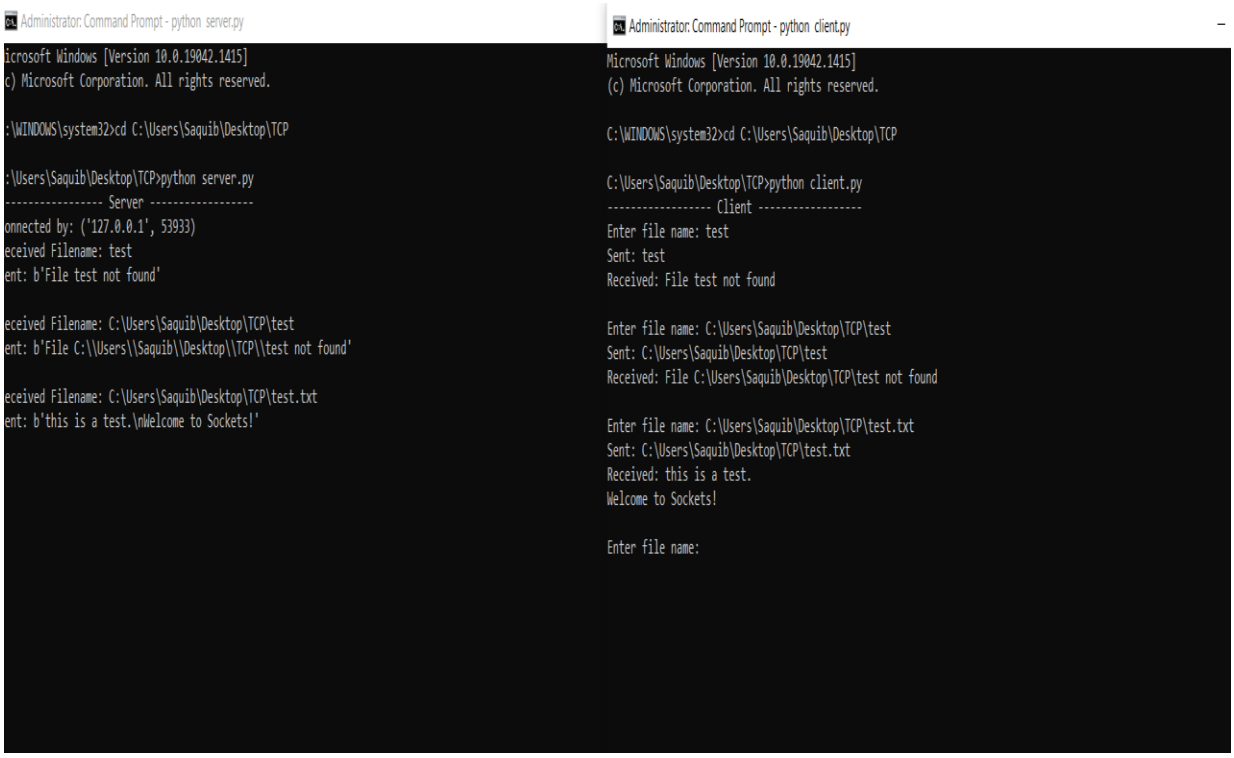
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
 serverSocket.listen(1)
while 1:
print("The server is ready to receive")
 connectionSocket, addr = serverSocket.accept()
sentence=connectionSocket.recv(1024).decode()
file = open(sentence, "r")
l = file.read(1024)
connectionSocket.send(l.encode())
 print('\nSent contents of ' + sentence) file.close()
connectionSocket.close()
```

Client:

```
from socket import *
serverName ='127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
 print(f"Recieved from {serverName}: ")
```

```
 filecontents = clientSocket.recv(1024).decode()
 print('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

OUTPUT:



6. Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if Present.

CODE:-

Server:

```
from socket import *
 serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
 serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
 file=open(sentence,"r")
l=file.read(2048)
serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '') file.close()
```

Client:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
```

```
sentence = input("\nEnter file name: ")
clientSocket.sendto(sentence.encode(), (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print('\nReply from Server:\n')
 print(filecontents.decode("utf-8"))
# for i in filecontents:
# print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```

OUTPUT: