PROGRAM 9 : TCP/IP-CLIENT/SERVER

CODE

PART 1 — SERVER.PY

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
print("Ready to recieve")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file.open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    file.close()
connectionSocket.close()
```

CODE

PART 2 — CLIENT.PY

```
from socket import *

serverName = "127.0.0.1"
serverPort = 12000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
```

```
def print_routing_table (self, node, dist, next_hop):
    print(f'Routing table for {node}:')
    print('Dest \t Cost \t Next Hop')
    for dest, cost in dist.items():
        print(f'{dest} \t {cost} \t
              {next_hop[dest]}')

def start(self):
    pass
```

PROGRAM 7B: DIJKSTRA'S ALGORITHM

CODE

```
import sys

class graph:

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

    def printSolution(self, dist):
        print("Vertex \t Distance from source")
        for node in range(self.V):
            print(node, "\t", dist[node])

    def minDistance(self, dist, sptSet):
        min = sys.maxsize
        for v in range(self.V):
            if dist[v] < min and sptSet[v] == False:
                min = dist[v]
```