**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB RECORD**

**Computer Network Lab (22CS4PCCON)**

***Submitted by***

**Chahat Singh (1BM22C075)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

****

**B.M.S. COLLEGE OF ENGINEERING**

# (Autonomous Institution under VTU)

**BENGALURU-560019**

**Academic Year 2024-25 (odd)**

B.M.S. College of Engineering

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**

****

**CERTIFICATE**

This is to certify that the Lab work entitled “ Computer Network (22CS4PCCON)” carried out by **Chahat Singh (1BM22CS075),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Dr. Kavitha Sooda Professor & HOD

Department of CSE, BMSCE

Nandhini Vineet

Assistant Professor

Department of CSE, BMSCE

**Index**

**CYCLE - 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.**  **No.** | **Date** | **Experiment Title** | **Page No.** |
| 1 | 9/10/24 | Simulate sending a simple PDU from source to destination using hub and switch as connecting devices | 1 - 5 |
| 2 | 16/10/24 | Configure IP address to routers in packet tracer | 6 – 9 |
| 3 | 23/10/24 | Configure default route, static route to the Router.(Part 1) | 10 – 14 |
| 4 | 13/11/24 | Configure default route, static route to the Router (Part 2) | 15 – 21 |
| 5 | 20/11/24 | Configure DHCP within a LAN and outside LAN | 22 – 27 |
| 6 | 27/11/24 | Configure RIP routing Protocol in Routers | 28 – 32 |
| 7 | 27/11/24 | Demonstrate the TTL/ Life of a Packet | 33 – 37 |
| 8 | 11/12/24 | Configure OSPF routing protocol | 38 – 45 |
| 9 | 18/12/24 | Configure Web Server, DNS within a LAN. | 46 – 48 |
| 10 | 18/12/24 | To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP) | 49 - 51 |
| 11 | 18/12/24 | To understand the operation of TELNET by accessing the router in server room from a PC in IT office. | 52 – 55 |
| 12 | 18/12/24 | To construct a WLAN and make the nodes communicate wirelessly | 56 – 58 |
| 13 | 18/12/24 | To construct a VLAN and make the PC’s communicate among a VLAN | 59 - 63 |

**CYCLE - 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Date** | **Experiment Title** | **Page No.** |
| 1 | 18/12/24 | Write a program for error detecting code using CRC-CCITT (16-bits). | 64 – 65 |
| 2 | 18/12/24 | Write a program for congestion control using Leaky bucket algorithm. | 65 – 69 |
| 3 | 18/12/24 | Write a client-server program using TCP/IP | 70 – 74 |
| 4 | 18/12/24 | Write a client-server program using UDP sockets | 75 - 79 |

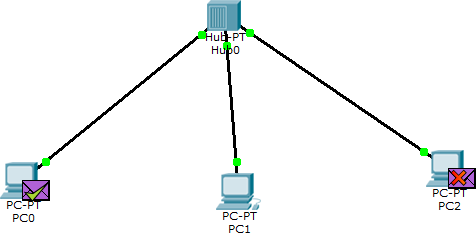
**Program 1**

# Aim of the program:

Simulate sending a simple PDU from source to destination using hub and switch as connecting devices

1. Using Hub

# Procedure along with the topology:

****

* 1. Select 3 end devices and a hub.
  2. Connect all the devices to the hub with a straight through cable.
  3. For each device do the configuration to set IP address for fast Ethernet.
  4. Add simple PDU, select source and destination and simulate the experiment.

# Observation:

* + - Source - PCO, Destination - PC2
    - Message is sent from PCO to hub.
    - The hub sends the message to both PC1 and PC2 devices. PC1 rejects while PC2 accepts the message.
    - PC2 sends feedback to the hub, hub sends it to PCO to PC1. PCO accepts while PC1 rejects the message.

# Using Switch

**Procedure along with the topology:**

****

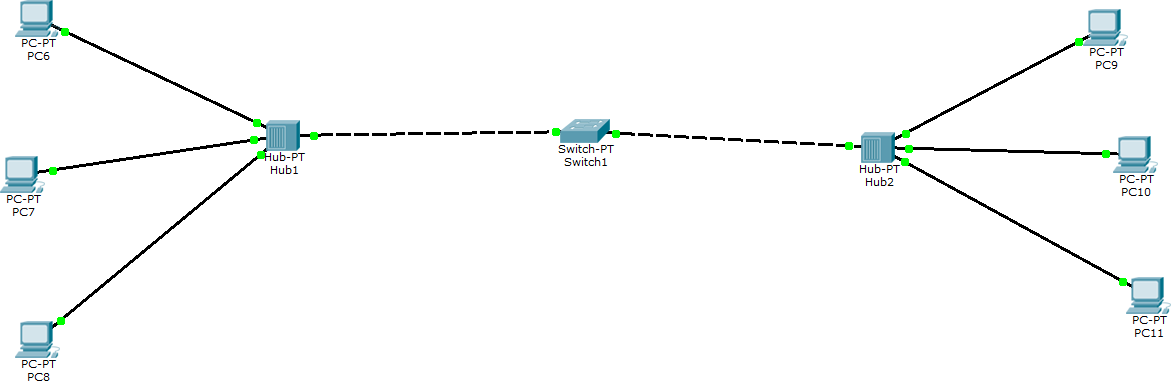
* 1. Select 4 end devices and a switch.
  2. Connect all the devices to the switch with straight through cable.
  3. For each device do the configuration. Set IP address for fast ethernet.
  4. Add simple PDU, select source and destination and simulate the experiment.

# Observation:

* + - Source: PCO , Destination: PC2
    - Message is sent from PCO to switch.
    - Switch sends the message to PC2.
    - Feedback is sent from to switch which then sends it to PCO.

# Using switch and hub

**Procedure along with the topology:**

****

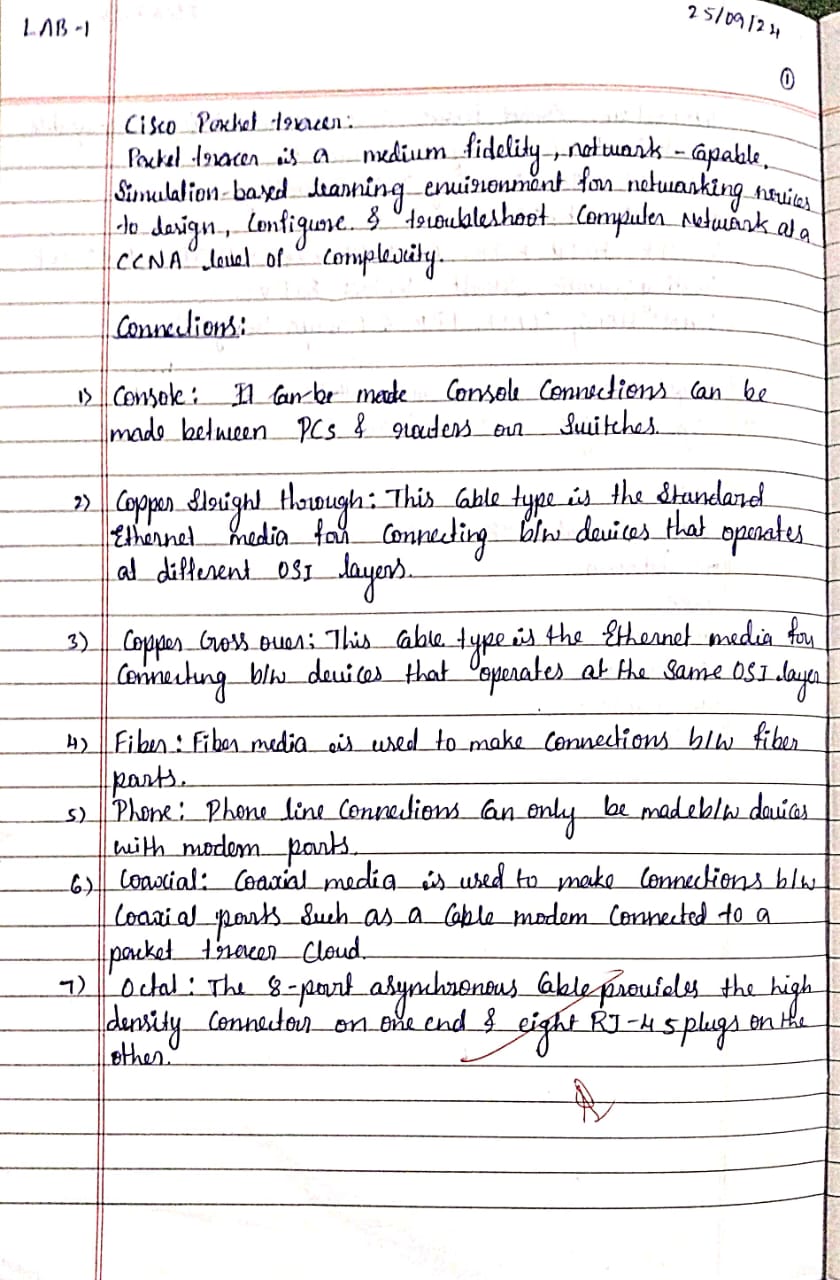
* 1. Select 1 switch, 2 hubs and 6 end devices.
  2. Connect first 3 devices to Hub 0 and other 3 devices to Hub 1, connect the two hubs to the switch.
  3. Do the configuration (set IP addresses) for each device.
  4. Add simple PDU, select source and destination and simulate the experiment.

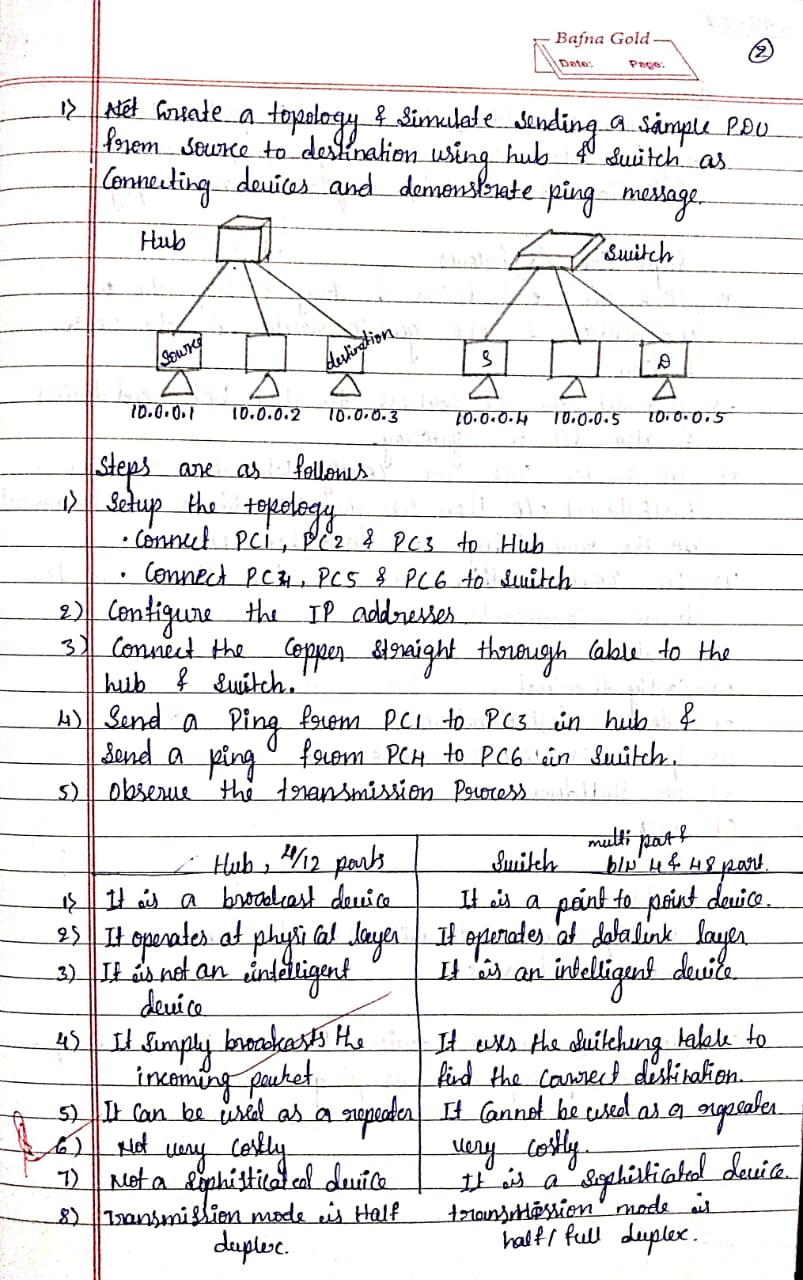
# Observation:

* + - Source: PCO , Destination: PC4
    - Message is sent from PCO to Hub 0.
    - Hub 0 sends message to PC1, PC2 and switch
    - Switch sends to Hub 1, Hub 1 sends to switch, PC3, PC4 and PC5. PC4 accepts while others reject.
    - Similarly, feedback is sent by PC4 to the PCO.

# 

# Observation Book:

****

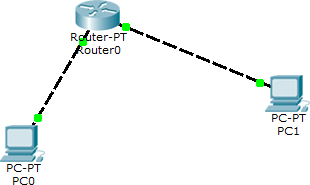
****

**Program 2**

# Aim of the program:

Configure IP address to routers in packet tracer.

# Procedure along with the topology:

****

Step 1

1. Place 2 generic PCs and one generic router.
2. Connect both PCs to the router's fast ethernet ports using copper crossover wire.

Step 2

1. Select PC1 config -> Fast Ethernet 0
2. Set IP address as 10.0.0.10 and default gateway 10.0.0.1.
3. Similarly for PC 2 IP 20.0.0.10 and default gateway 20.0.0.1.

Step 3

1. Select router and go to CLI
2. Execute the following commands
   * >enable
   * #config terminal
   * # interface Fast Ethernet 0/0
   * # ip address 10.0.0.1 255.0.0.0
   * # no shutdown
   * # exit

PC 0 and router are successfully connected

Now run for PC 1

* + interface Fast Ethernet 1/0
  + # ip address 20.0.0.1 255.0.0.0
  + # no shutdown Successfully connected

Step 4: Select the PC 0 and open the command prompt.

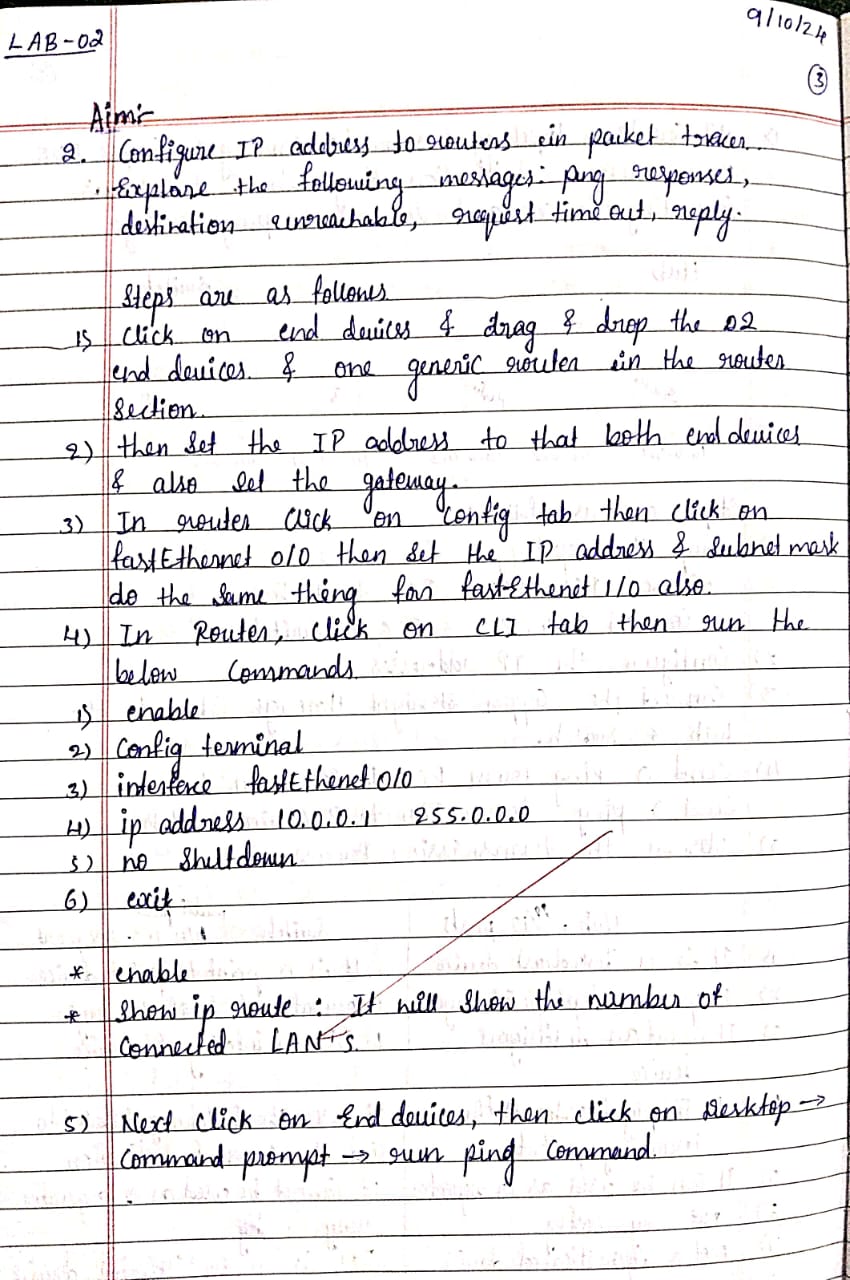
Step 5: Ping the PC 1 using the following command: ping 20.0.0.10 Step 6: Observe the output.

# Screen shots/ output :

**Observation:**

PC 0 succesfully pings with PC1 with 32 bytes of data.

**Observation Book:**

****

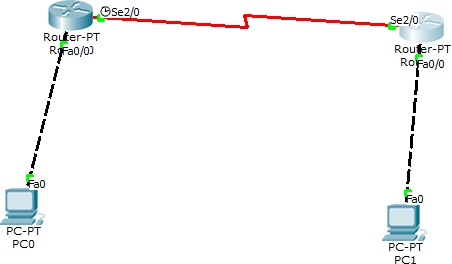


**Program 3**

# Aim of the program:

Configure default route, static route to the Router.(Part 1)

# Procedure along with the topology:

****

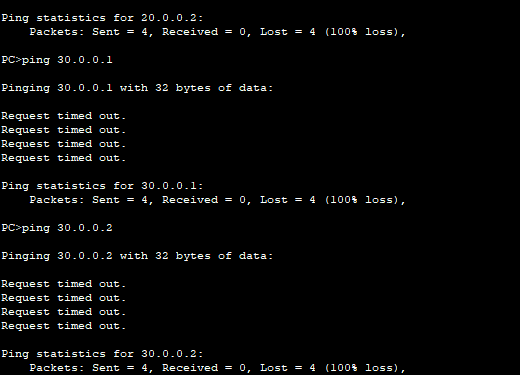
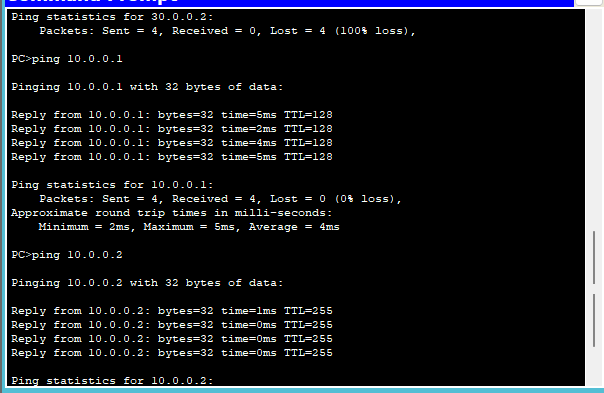
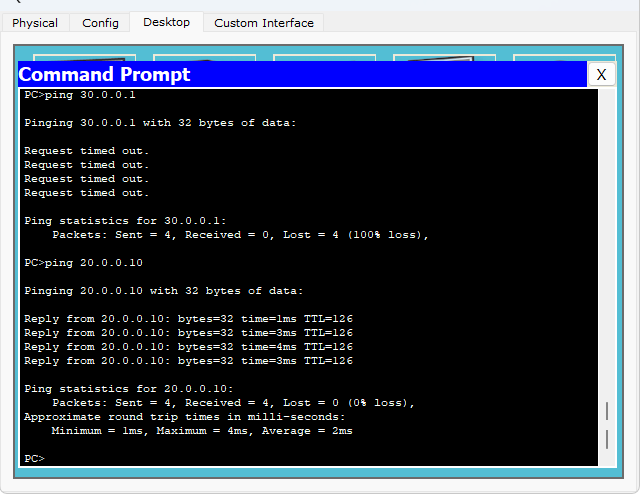
1. Select a generic router R1.
2. Connect an end device PC1 to router R1 through parallel connection - fastethernet 010.
3. Configure PC1 with ip address 10.0.0.10 and gateway 10.0.0.2
4. Similarly select another generic router R2 and connect an end device PC2 - fastethernet 010.
5. Configure PC2 with ip address 20.0.0.10 and gateway 20.0.0.2.
6. Now, select router R1, go to CLI and execute the following.
   * Router> enable
   * Router# config terminal
   * Router(config)# interface fastethernet 0/0
   * Router(config-if)# ip address 10.0.0.1 255.255.255.0
   * Router(config-if)# no shutdown "Interface fastethernet 0/0, changed state to up"
7. Similarly select routes R2 goto (1) and execute the same
   * Router> enable
   * Router# config terminal
   * Router(config)# interface fastethernet 1/0
   * Router(config-if)# ip address 20.0.0.2 255.255.255.0
   * Router(config-if)# no shutdown

“Interface fastethernet 1/0, changed state to up”

1. Hence the connection between Router and end devices is established.
2. Now connect router R1 with router R2 using serial cable (specially connected)
3. To setup connection b/w routers again
   * Select router R1 and go to CLI
   * Router(config)# interface serial 2/0
   * Router(config-if)# ip address 30.0.0.1 255.0.0.0
   * Router(config-if)# no shutdown
   * Select router R2 and go to CLI
   * Router(config)# interface serial 3/0
   * Router(config-if)# ip address 30.0.0.2 255.0.0.0
   * Router(config-if)# no shutdown

“Interface serial 2/0 changed state to up”

# Screen shots/ output :

****

**Observations:**

* After setting up the above mentioned topology ,
* Now try to ping PC1 with PC0
* Open command prompt for PC1 and type -> ping 20.0.0.1
* Destination net host unreachable
* Packets sent = 4, received = 0, lost = 4, loss = 100%

It is also observed that the end system PC0 was only pinged with router R0 only

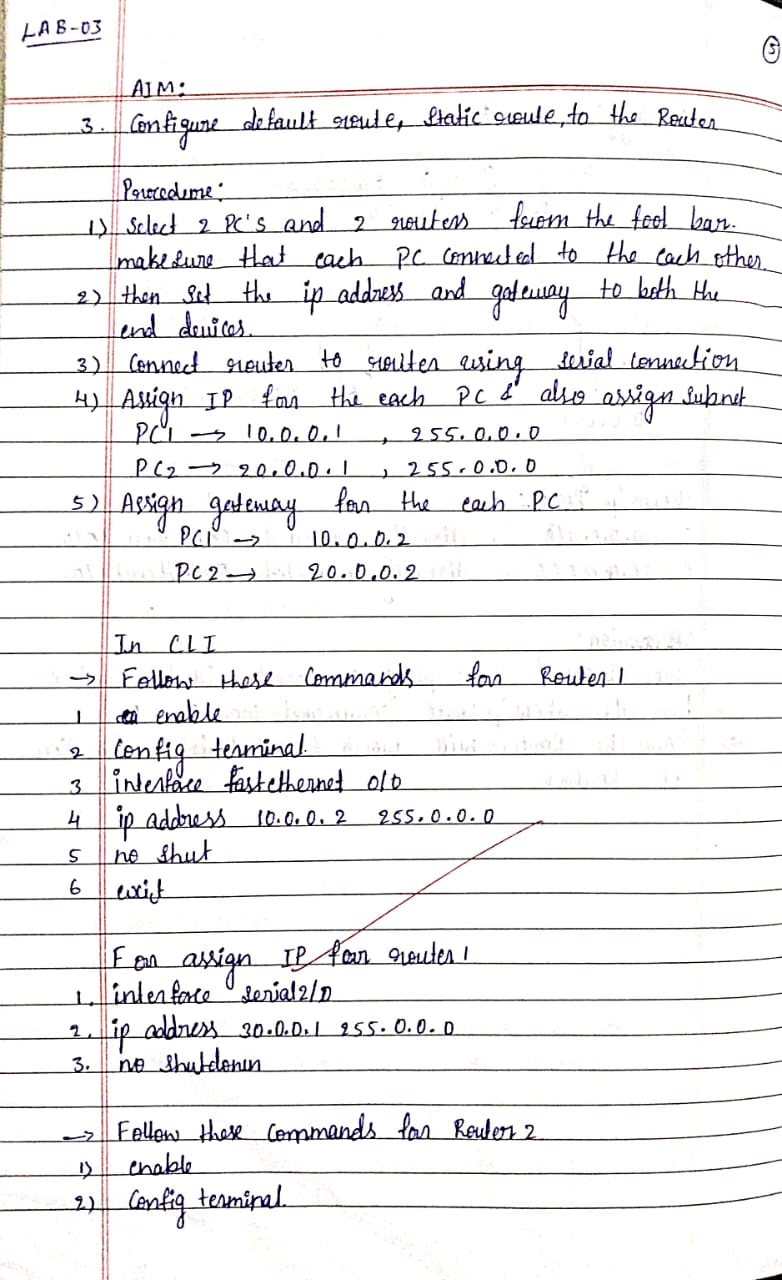
In order to establish static connection ,go to router 0's CLI and type the following commands

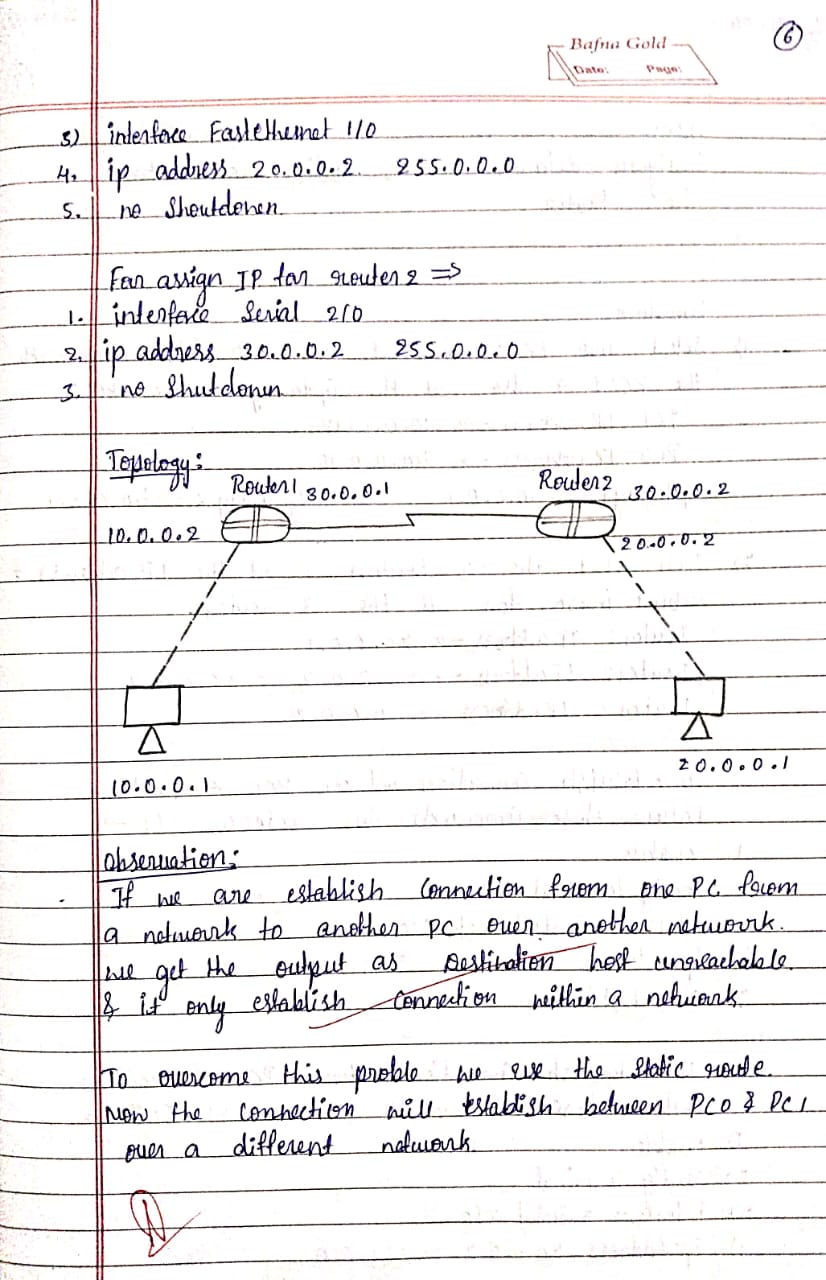
* + Router> enable
  + Router# config terminal
  + Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.2 Then go to router 2's CLI and type the following commands
  + Router> enable
  + Router# config terminal
  + Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1

So now check if all the connections are done by going to each router's CLI and typing show ip route.

* After this again go to PC1 and now type in CP
* ping 20.0.0.10
* Packets: sent = 4, Received = 4, Lost = 0

**Observation Book:**

****

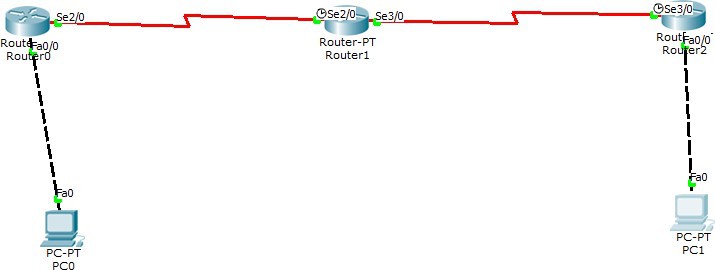
****

**Program 4**

# Aim of the program:

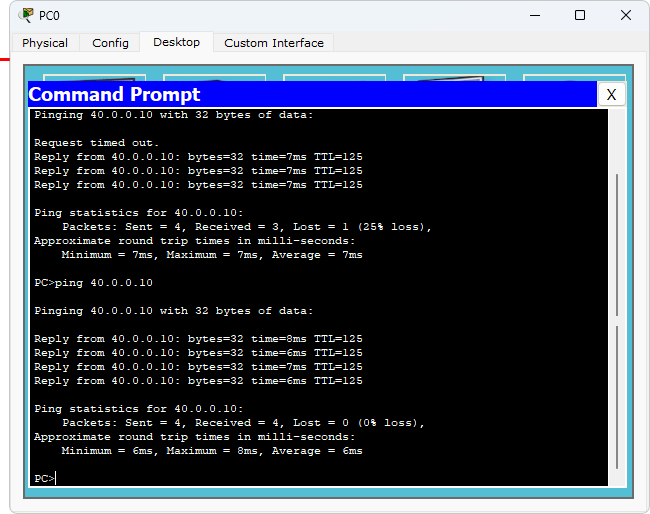
Configure default route, static route to the Router.(Part 2)

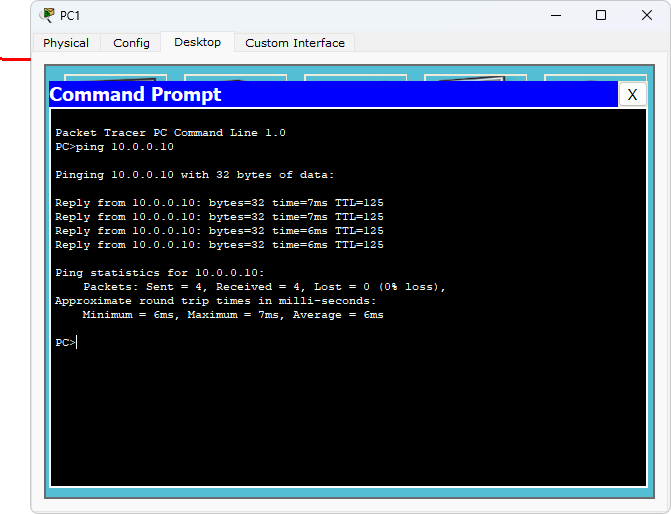
# Procedure along with the topology:

****

1. Select a generic router R0
2. Connect an end device PC0 to router R0 through parallel connection fast-ethernet 0/0
3. Configure PC0 with ip address 10.0.0.10 and gateway 10.0.0.1
4. Select another router R1
5. There is a serial connection from router R0 to R1 i.e Se2/0
6. Select another router R2 and there is a serial connection from router R1 to R2 i.e Se3/0
7. From Router R2 there connect a PC1 through parallel connection fa0/0
8. Configure PC1 with ip address 40.0.0.10 and gateway 40.0.0.1
9. Now go to Router 0 and CLI and type
   * Router> enable
   * Router# config terminal
   * Router(config)# interface fastethernet0/0
   * Router(config-if)# ip address 10.0.0.1 255.0.0.0
   * Router(config-if)# no shutdown "Interface FastEthernet 0/0 changed state to up"
10. Similarly go to Router 1 and CLI and type
    * Router> enable
    * Router# config terminal
    * Router(config)# interface serial2/0
    * Router(config-if)# ip address 20.0.0.2 255.0.0.0
    * Router(config-if)# no shutdown "Interface Serial 2/0 changed state to up"
11. So now we have to do for serial3/0
    * Router> enable
    * Router# config terminal
    * Router(config)# interface serial3/0
    * Router(config-if)# ip address 30.0.0.1 255.0.0.0
    * Router(config-if)# no shutdown "Interface Serial 3/0 changed state to up"
12. Now go to Router 2 and CLI and type
    * Router> enable
    * Router# config terminal
    * Router(config)# interface fastethernet0/0
    * Router(config-if)# ip address 40.0.0.1 255.0.0.0
    * Router(config-if)# no shutdown "Interface FastEthernet 0/0 changed state to up"
13. So now we will do for serial 3/0
    * Router(config)# interface serial3/0
    * Router(config-if)# ip address 30.0.0.2 255.0.0.0
    * Router(config-if)# no shutdown
14. Now we will set static Route
15. So for that we will go to Router 1 CLI and type
    * Router# show ip route
    * C 20.0.0.0/8 is directly connected, Serial2/0
    * C 30.0.0.0/8 is directly connected, Serial3/0
16. So now again type
    * Router(config)# terminal
    * Router(config)# ip address route 10.0.0.0 255.0.0.0 20.0.0.1
    * Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2
    * Router(config)# exit
    * Router# show ip route
    * S 10.0.0.0/8 via 20.0.0.1
    * C 20.0.0.0/8 is directly connected, Serial2/0
    * C 30.0.0.0/8 is directly connected, Serial3/0
    * S 40.0.0.0/8 via 30.0.0.1
17. Now we have to set default Route
18. So we will go to Router 0
    * Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
19. Then Router 2
    * Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1"

# Screen shots/ output :

****



**Observation:**

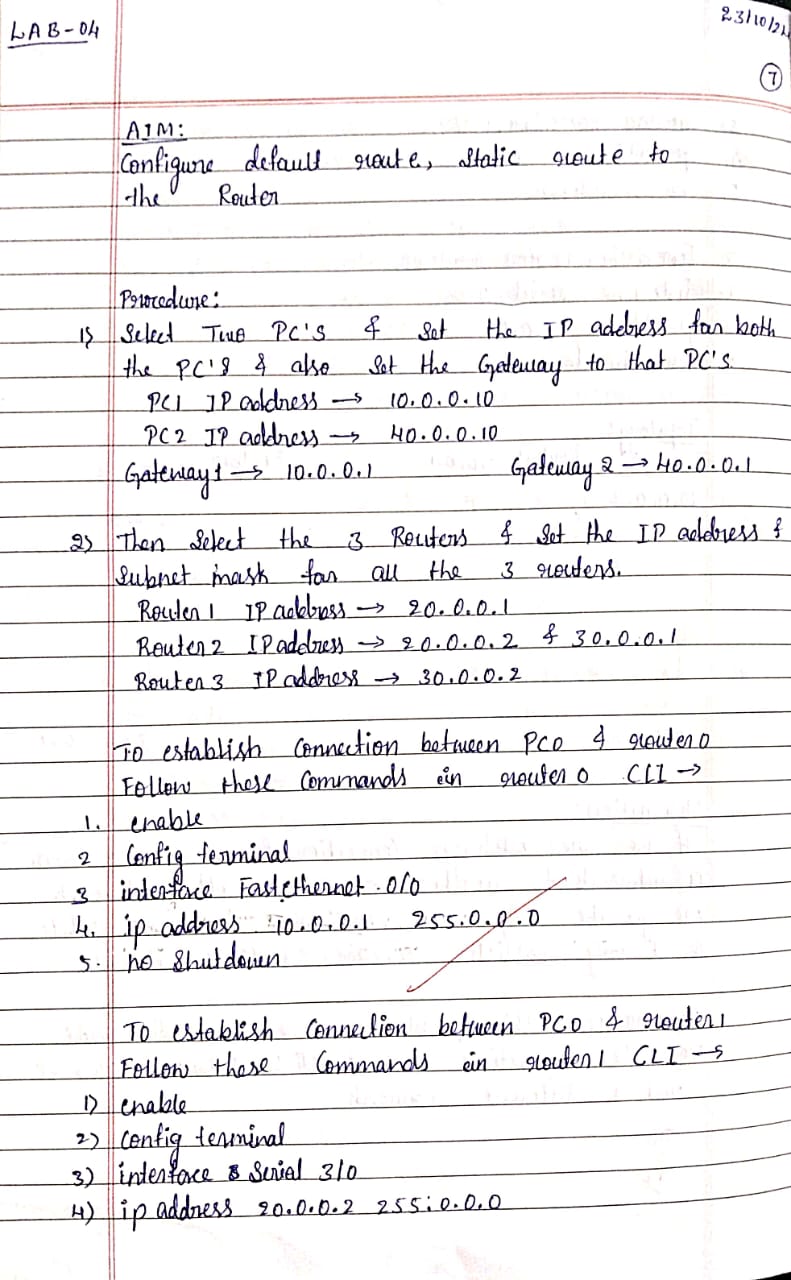
So now we will go to PC0 and in the command prompt type:

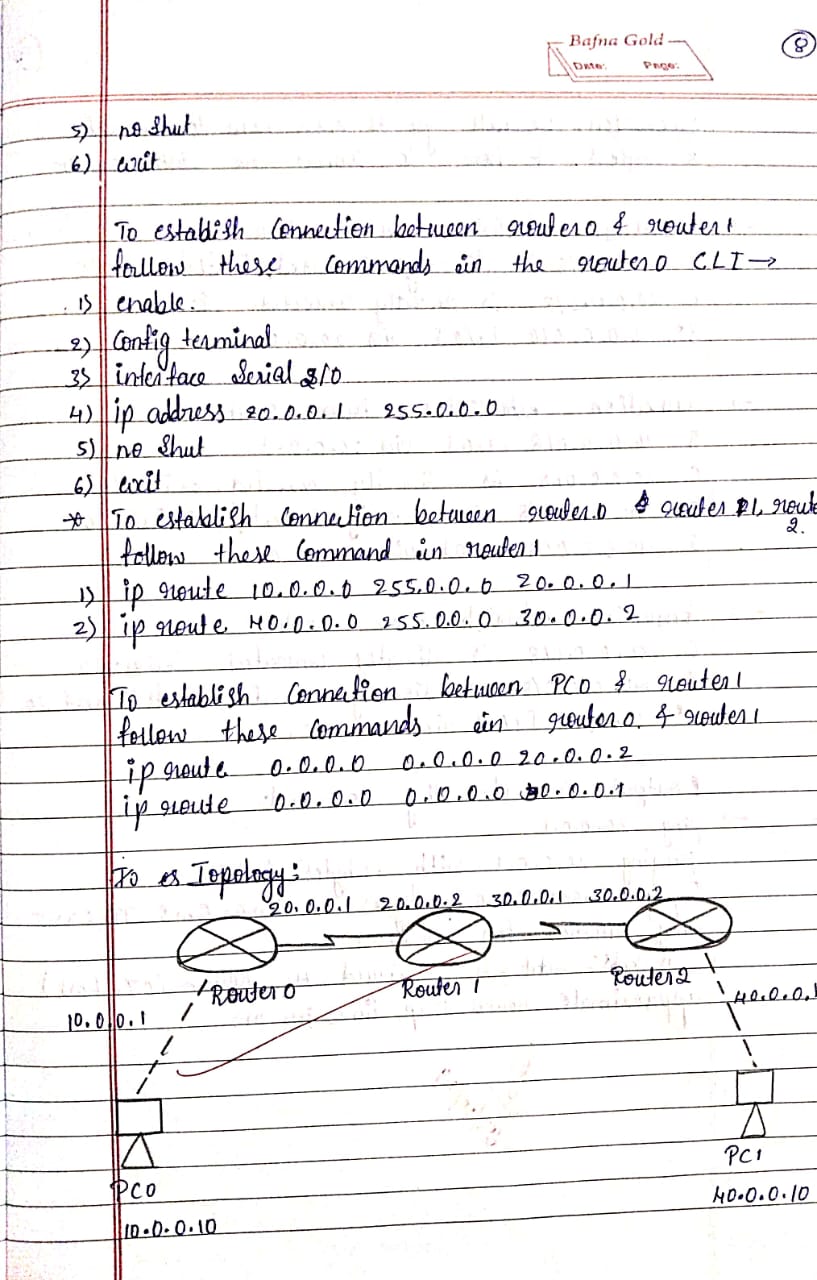
* PC> ping 40.0.0.10
* Pinging 40.0.0.10 with 32 bytes of data
* Ping statistics for 40.0.0.10
* Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

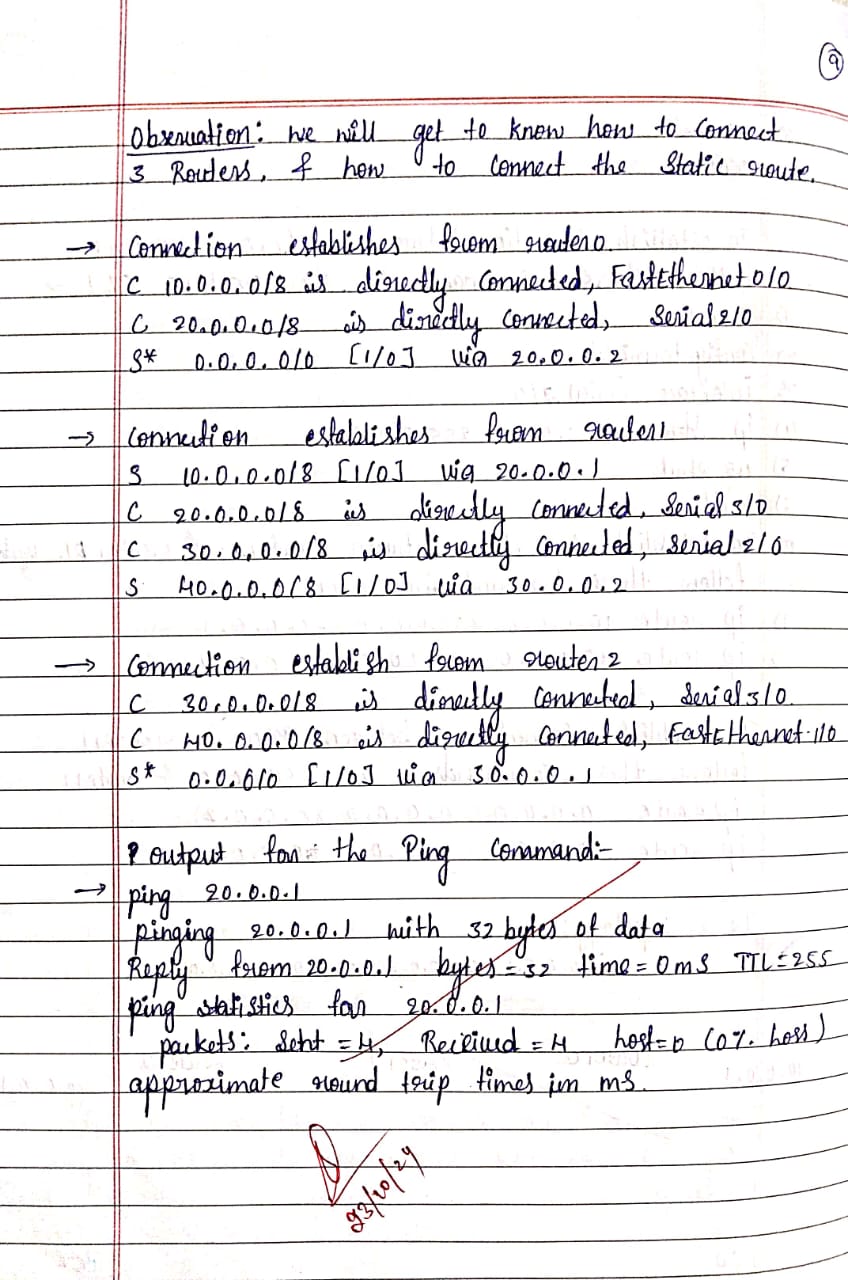
Now to ping PC1 we have to go to command prompt and type:

* PC> ping 10.0.0.10
* Pinging 10.0.0.10 with 32 bytes Packets: Sent = 4, Received = 4, Lost = 0

**Observation Book:**







**Program 5**

# Aim of the program:

Configure DHCP within a LAN and outside LAN

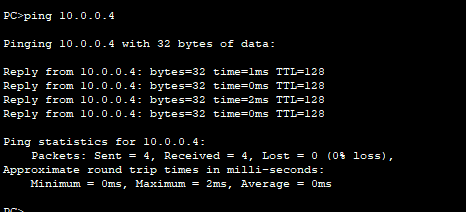
1. Within LAN

# Procedure along with the topology:

****

* 1. Select generic server-PT
  2. Select generic PC0, PC1, laptop-PT laptop0, Switch, laptop-1, laptop-2, laptop-3 and a router-0.
  3. All are connected used Automatically choose Connection Type.
  4. Click on server-PT and give the
  5. IP address: 10.0.0.1
  6. Subnet Mask: 255.0.0.0
  7. Default Gateway: 10.0.0.10
  8. Select DHCP and set service as on. Set the pool Name as switch one, default gateway as 10.0.0.1, maximum number of users as 100 and start IP address as 10.0.0.3. Click Add.
  9. Click on each PC, navigate to IP configuration and select DHCP.
  10. Ping from one device to another

# Screen shots/ output :

****

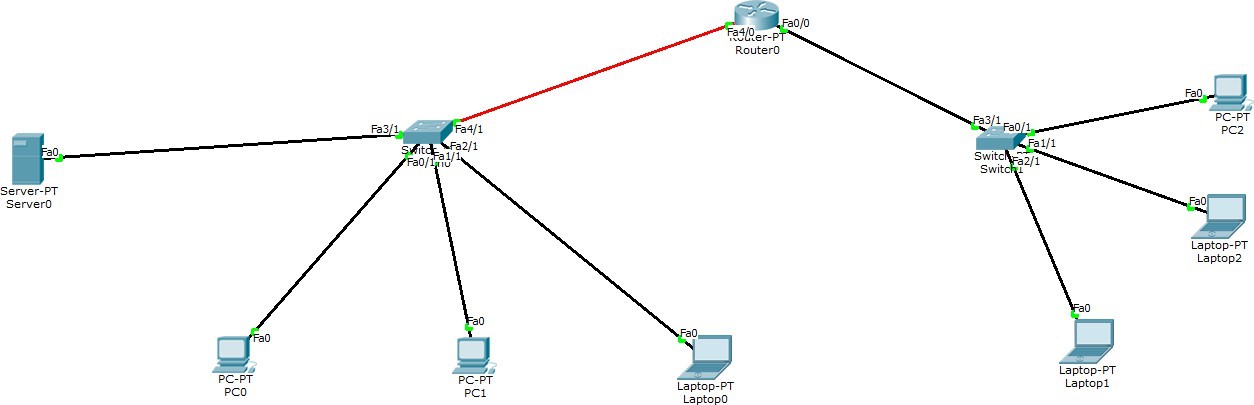
**Observation:**

Pinging 10.0.0.4 with 32 bytes of data

Reply from 10.0.0.4: bytes=32 time=1ms TTL=128 Reply from 10.0.0.4: bytes=32 time=0ms TTL=128 Reply from 10.0.0.4: bytes=32 time=0ms TTL=128 Reply from 10.0.0.4: bytes=32 time=0ms TTL=128 Packets: Sent = 4, Received = 4, Loss = 0 (0% loss)

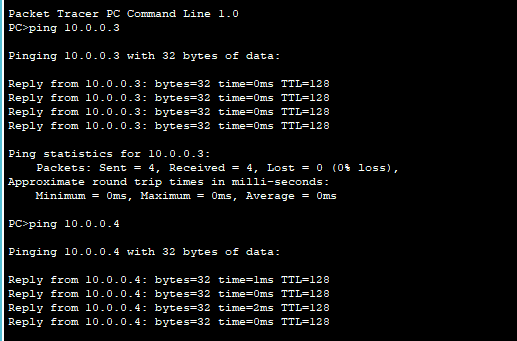
1. Outside LAN

# Procedure along with the topology:

****

* 1. Select server, set IP Address as 10.0.0.2, default gateway as 10.0.0.0, DNS server as 0.0.0.0
  2. Navigate into config, select DHCP, modify default gateway as 10.0.0.1, start IP address as 10.0.0.0 (switch 1)
  3. Name a new pool as switch two, set start IP as 20.0.0.3, default gateway as 20.0.0.1, maximum number of users as 100, set gateway in setting as 10.10.0.1
  4. Select router, navigate to CLI and enter the following commands:
     + enable
     + config terminal
     + interface fastethernet 0/0
     + ip address 10.0.0.1 255.0.0.0
     + ip helper address 10.0.0.2
     + no shutdown
     + exit
  5. Repeat the same

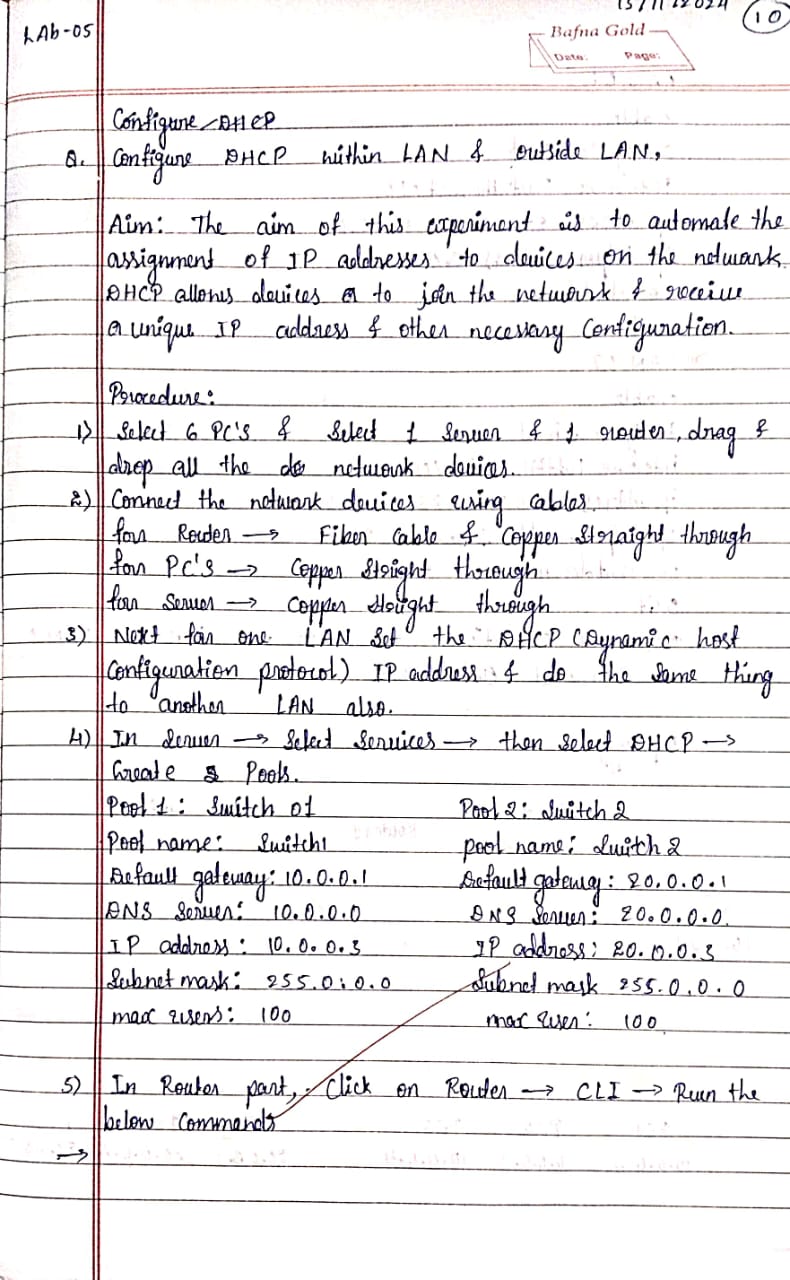
# Screen shots/ output :

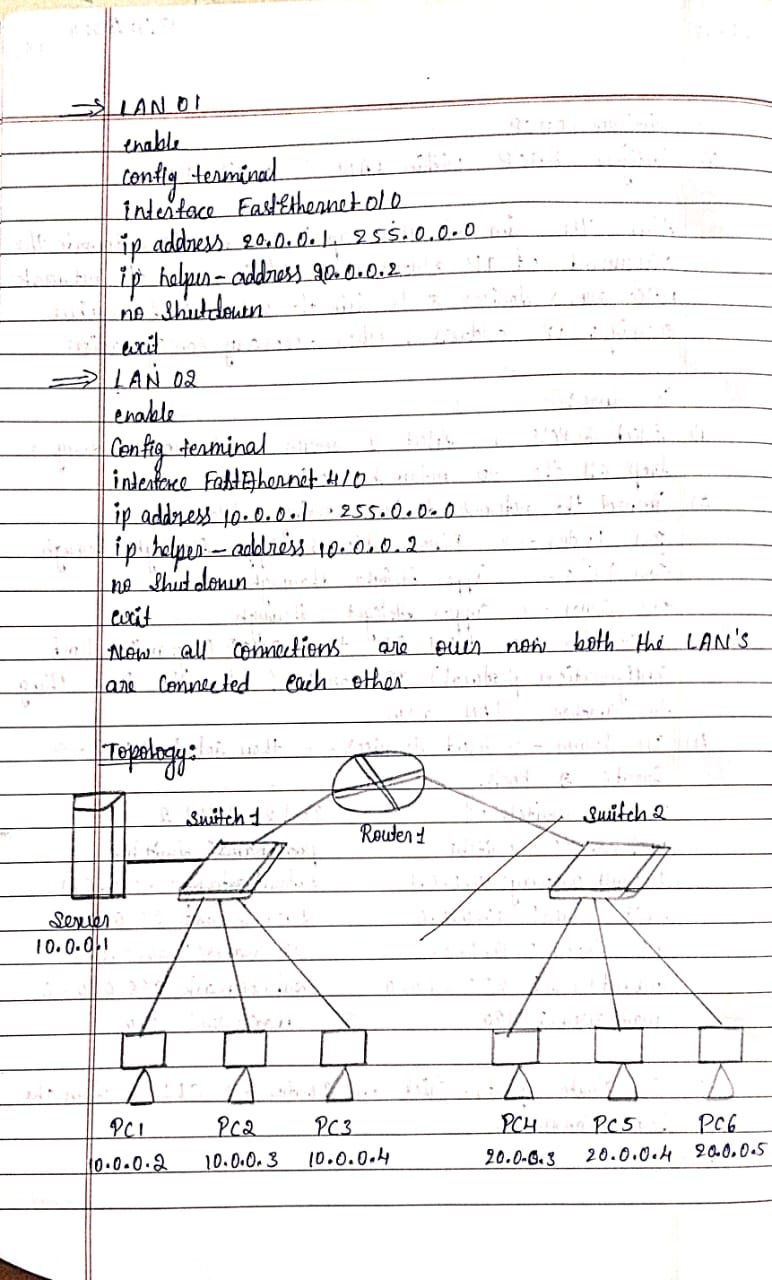
****

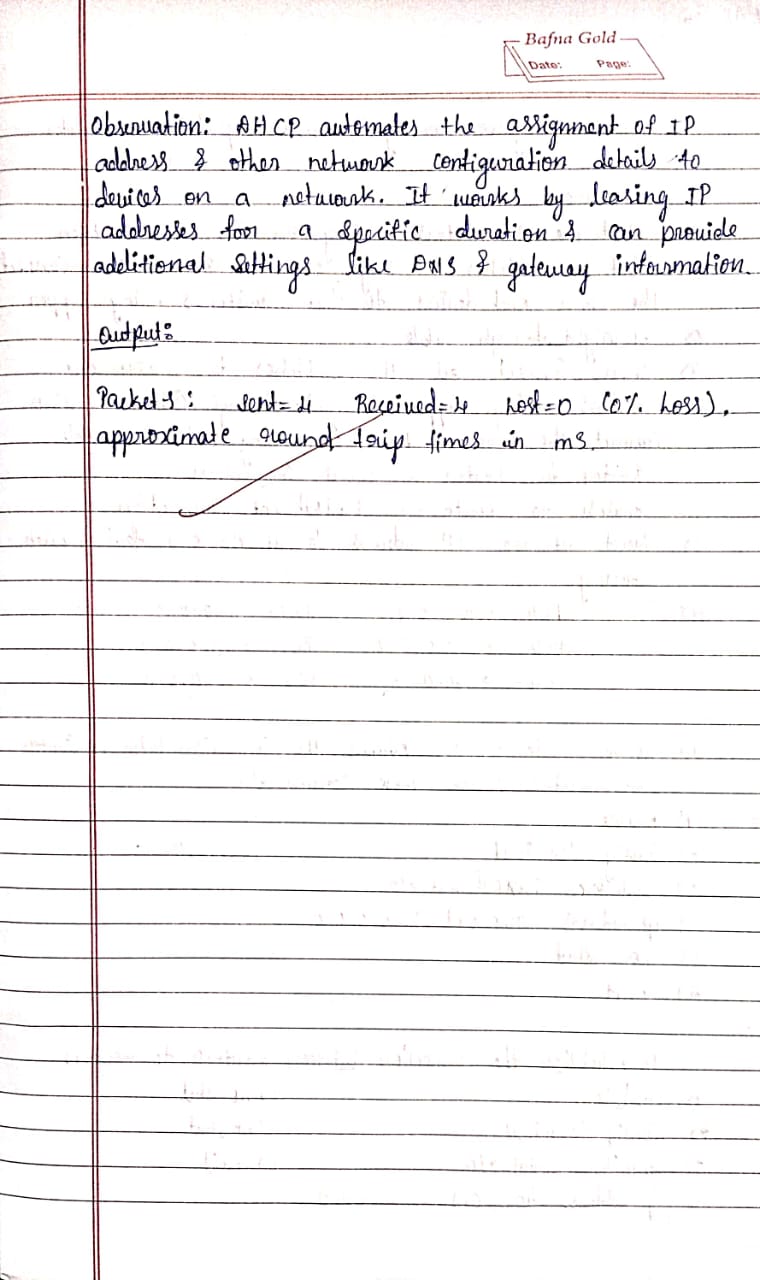
**Observation:**

The ip address is set for all end devices.

**Observation Book:**







**Program 6**

# Aim of the program:

Configure RIP routing Protocol in Routers

# Procedure along with the topology:

****

1. Configure the routers, switches and generic PCs
2. For the PCs, navigate to config interface and fastethernet 0 and set the IP address for all six PCs
3. Similarly for all six PCs, navigate to config, global settings and set gateway as the IP address of the router it is connected to
4. In router 0, navigate to config, fast ethernet 0/0 and add IP address (10.0.0.1)
5. Then go to serial 3/0 and update IP address as 40.0.0.1
6. Go to CLI and enter command no shut
7. In Router 1, Go to
   * Interface fast ethernet 0/0, set Ip address as 20.0.0.1.
   * Go to serial 2/0 and set IP address to 40.0.0.2
   * In CLI enter no shut.
   * Go to serial 3/0 and set Ip address as 50.0.0.1
   * In CLI enter no shut.
8. In Router 2 Go to Interface
   * fast ethernet 0/0, set IP address as 30.0.0.1
   * Go to serial 2/0 and set IP address as 50.0.0.2
9. If connections are not active after this, in all routers check the fast ethernet 0/0 and enter no shut in the CLI again. Now all the connections are active (green).
10. So now go to router 0 and in CLI type
    * Router(config)# router rip
    * Router(config-route)# network 10.0.0.0
    * Router(config-route)# network 40.0.0.0
    * Router(config-route)# exit
    * Router(config)# exit
    * Router# show ip route C ………..

R ………….. C ………….

1. Now go to router 1 and in CLI type
   * Router(config)# router rip
   * Router(config-route)# network 40.0.0.0
   * Router(config-route)# network 50.0.0.0
   * Router(config-route)# network 20.0.0.0
2. Router# show ip route

R

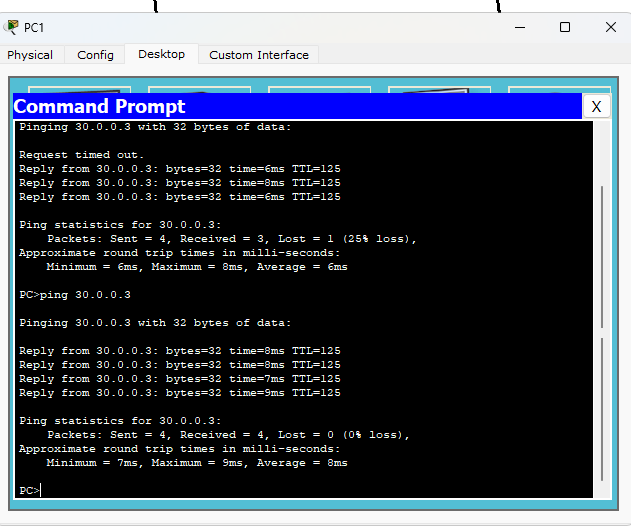
C . . . C

C . . .

1. Now go to router 2 and type
   * Router(config)# router rip
   * Router(config-router)# network 50.0.0.0
   * Router(config-router)# network 30.0.0.0
   * Router# show ip route R . . .

C

# Screen shots/ output :

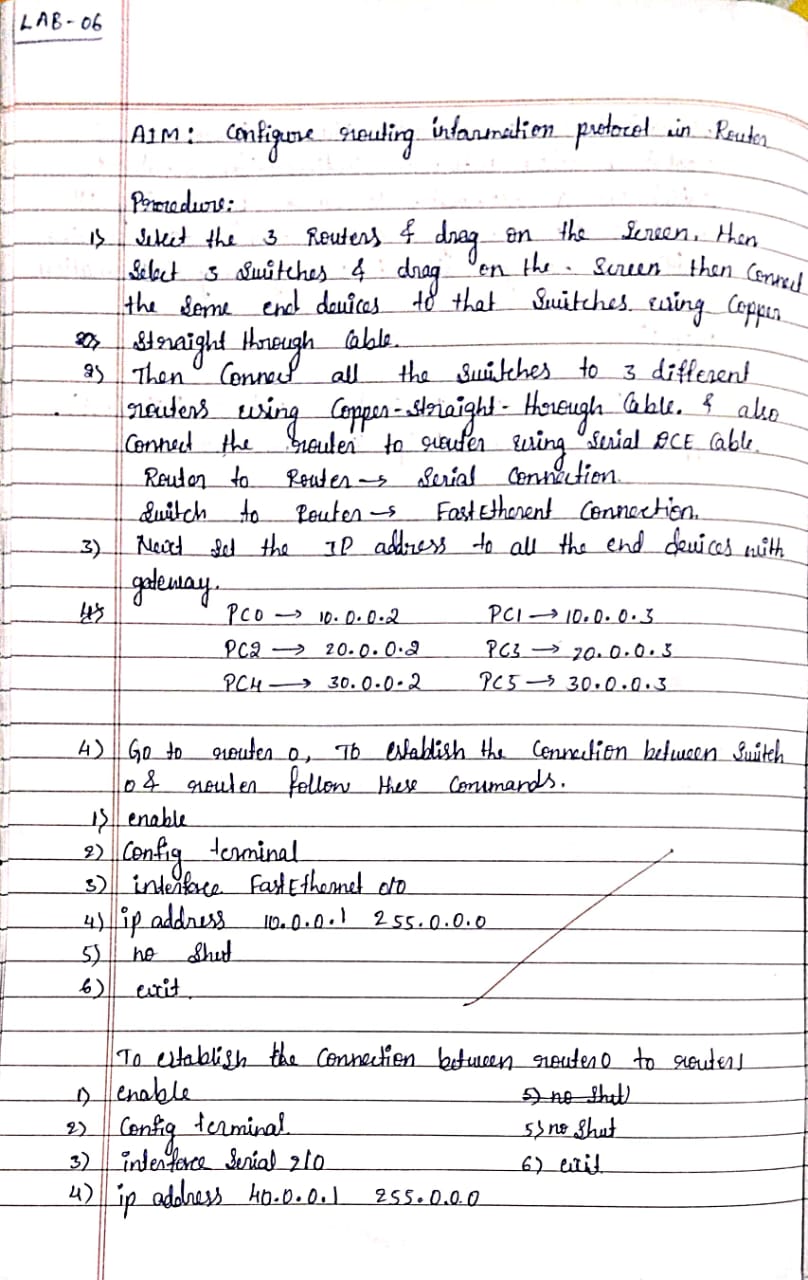
****

**Observation:**

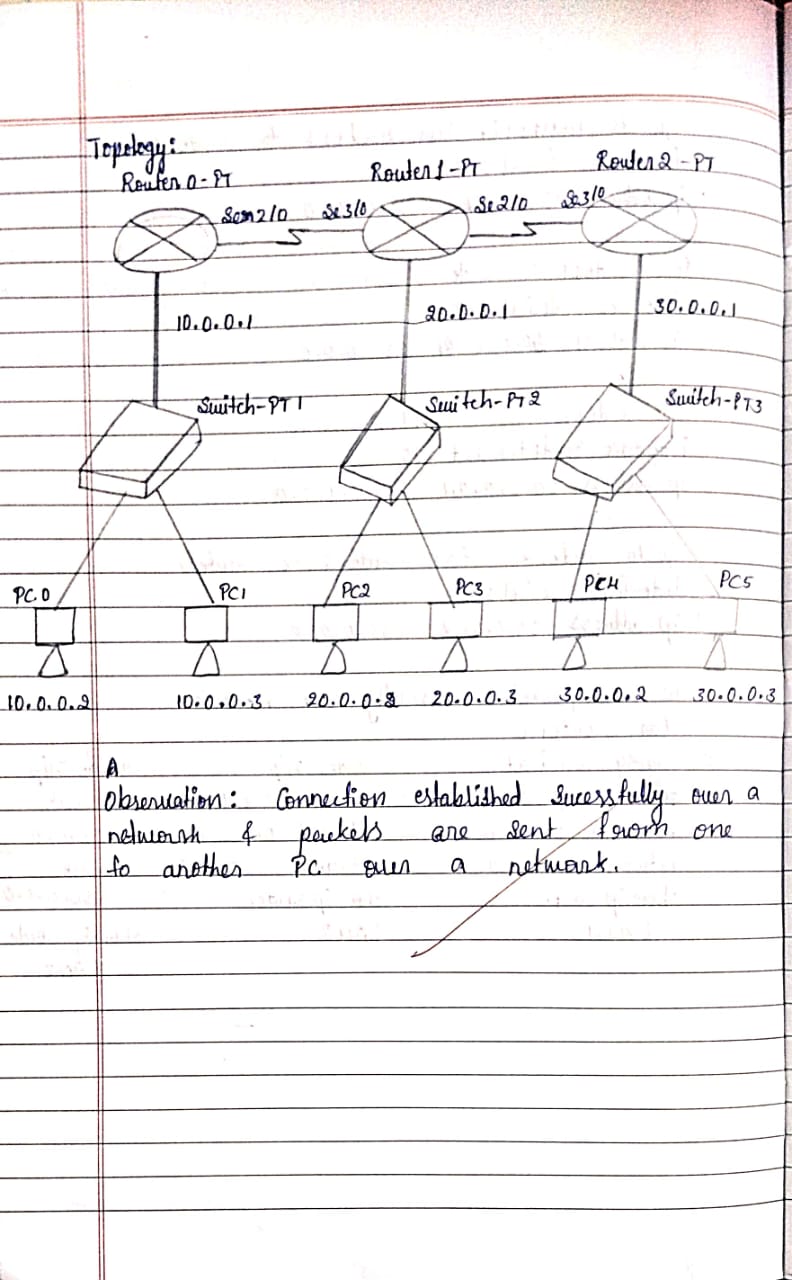
From PC1 we can ping to PC5 PC1> ping 30.0.0.3

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

**Observation Book:**

****

****

****

**Program 7**

# Aim of the program:

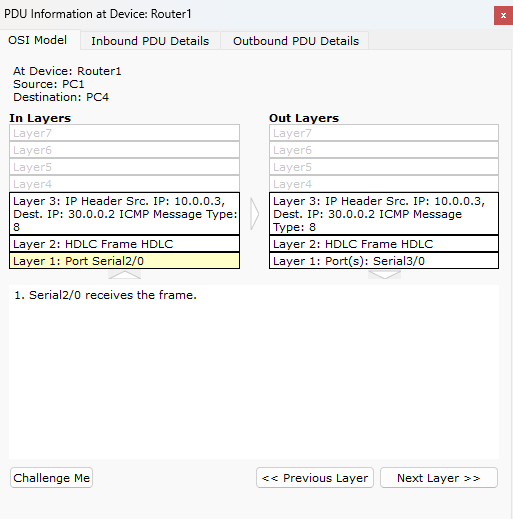
Demonstrate the TTL/ Life of a Packet:

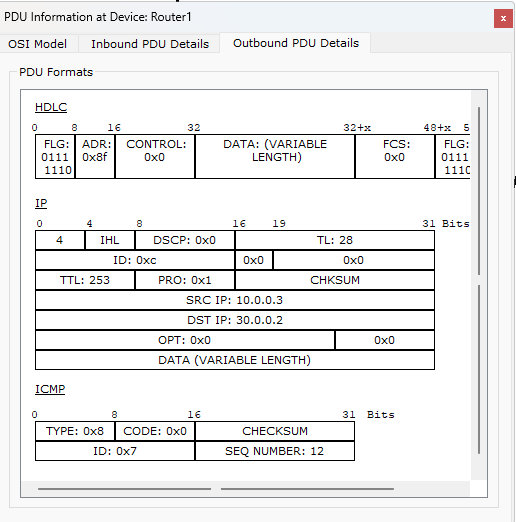
# Procedure along with the topology:

****

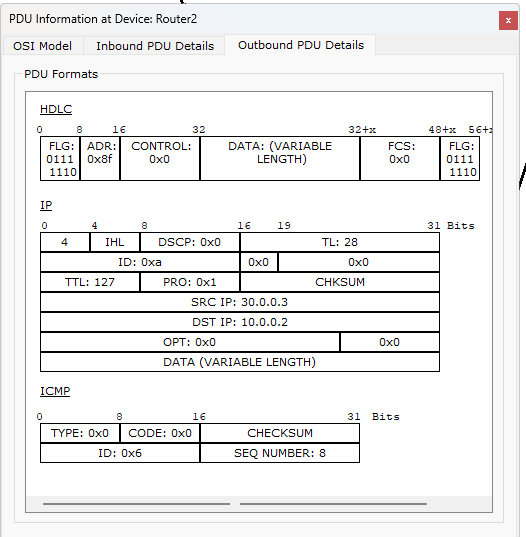
1. Add a simple PDU across the PCs of different network.
2. Consider PC0 to PC5

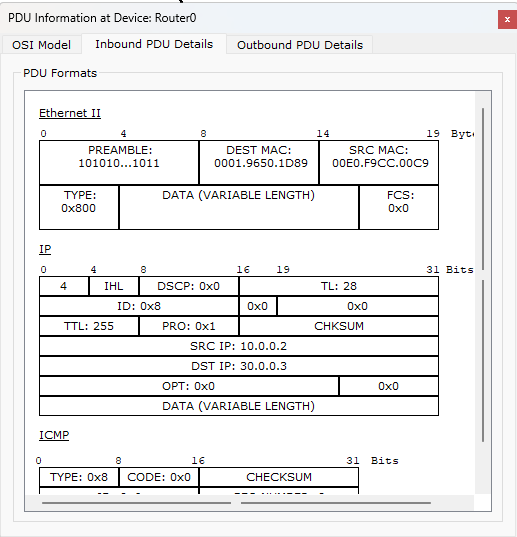
# Screen shots/ output :

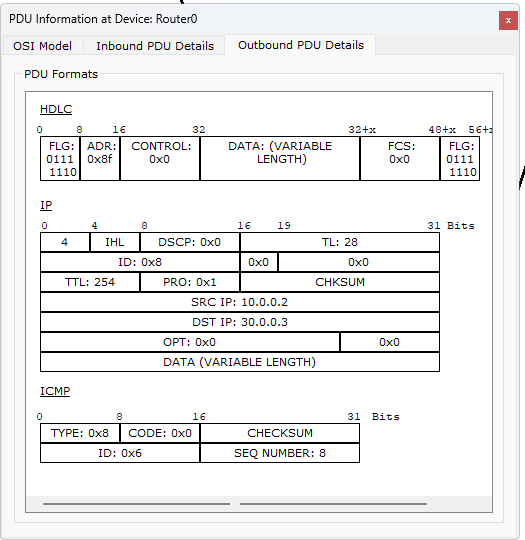
****



****

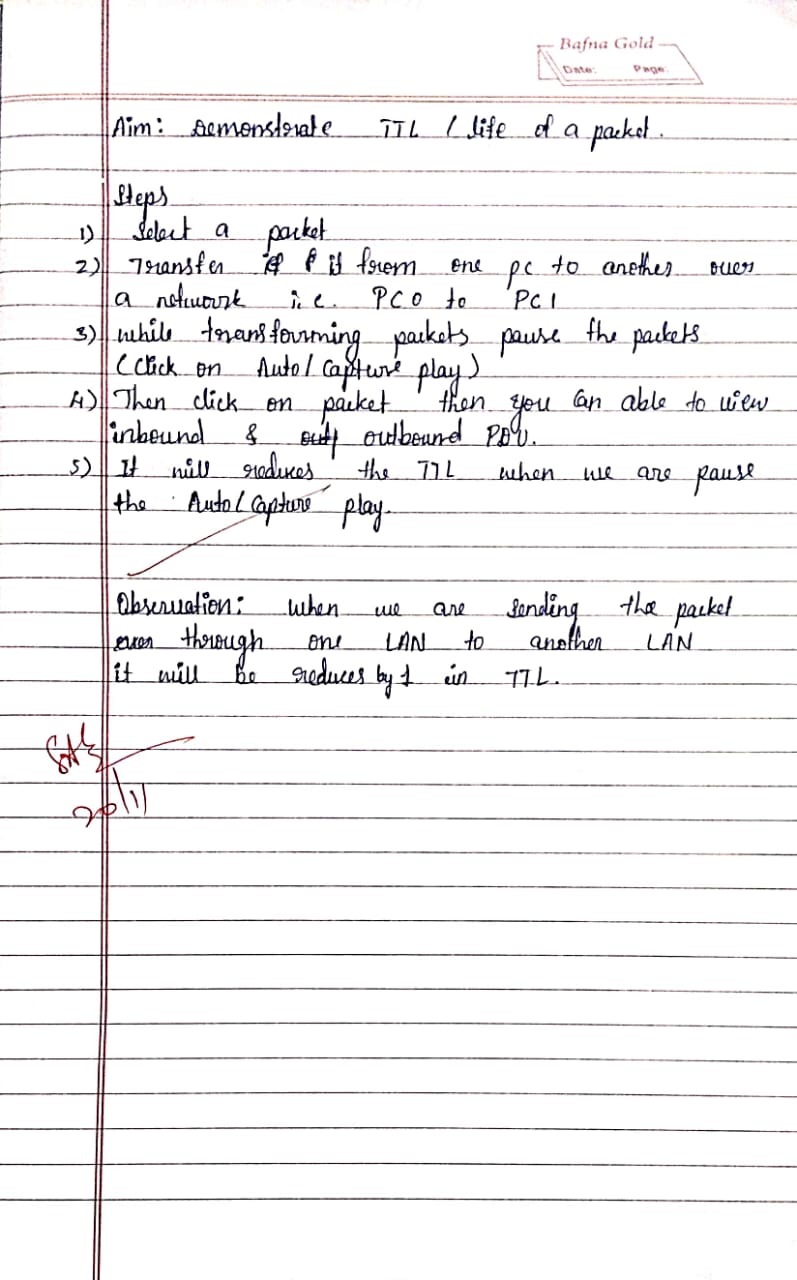


****



**Observations:**

* + While Auto capture and observing the TTL across each PC, it was observed as follows:
  + PDU Information at Device: PC1 TTL: 225
  + PDU Information at Device: Router1 TTL: 254
  + PDU Information at Device: Router2 TTL: 253
  + Cisco packet tracer has the maximum TTL as 225.
  + It is observed that the TTL document on the message is being passed stop by stop (router to router).
  + The figure of the OSI model of switch demonstrates the flow of packets in 2 layers while 3 layers in the router. The TTL reaches zero once all the packets are received.

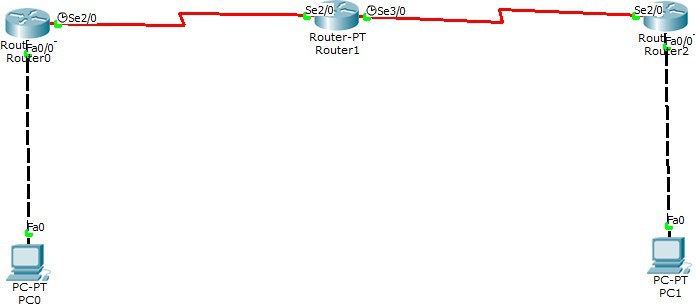
****

**Program 8**

# Aim of the program:

Configure OSPF routing protocol

# Procedure along with the topology:

****

1. Create topology like below i have given
2. Configure ip address to all interfaces
3. In Router RI,
   * Rl(config)#interface fastethernet 2/0
   * R 1( config-if)#ip address 10.0.0.1 255.0.0.0
   * Rl(config-if)#no shutdown
   * Rl(config-if)#exit
   * Rl(config)#interface serial 1/0
   * Rl(config-if)#ip address 20.0.0.1 255.0.0.0
   * Rl(config-if)#encapsulation ppp
   * Rl(config-if)#clock rate 64000
   * Rl(config-if)#no shutdown
   * Rl ( config-if)#exit
4. In Router R2
   * R2(config)#interface serial 1/0
   * R2(config-if)#ip address 20.0.0.2 255.0.0.0
   * R2(config-if)#encapsulation ppp
   * R2(config-if)#no shutdown
   * R2(config-if)#exit
   * R2(config)#interface serial 1/1
   * R2(config-if)#ip address 30.0.0.1 255.0.0.0
   * R2(config-if)#encapsulation ppp
   * R2(config-if)#clock rate 64000
   * R2(config-if)#no shutdown
   * R2(config-if)#exit
5. In Router R3
   * R3(config)#interface serial 1/0
   * R3(config-if)#ip address 30.0.0.2 255.0.0.0
   * R3(config-if)#encapsulation ppp
   * R3(config-if)#no shutdown
   * R3(config-if)#exit
   * R3(config)#interface fastethernet 2/0
   * R3(config-if)#ip address 40.0.0.1 255.0.0.0
   * R3(config-if)#no shutdown
   * R3(config-if)#exit
6. Configure OSPF
7. On Router R1:
   * R1(config)#router ospf 1
   * R1(config-router)#router-id 1.1.1.1
   * R1(config-router)#network 10.0.0.0 0.255.255.255 area 3
   * R1(config-router)#network 20.0.0.0 0.255.255.255 area 1
   * R1(config-router)#exit
8. On Router R2:
   * R2(config)#router ospf 1
   * R2(config-router)#router-id 2.2.2.2
   * R2(config-router)#network 20.0.0.0 0.255.255.255 area 1
   * R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
   * R2(config-router)#exit
9. On Router R3
   * R3(config)#router ospf 1
   * R3(config-router)#router-id 3.3.3.3
   * R3(config-router)#network 30.0.0.0 0.255.255.255 area 0
   * R3(config-router)#network 40.0.0.0 0.255.255.255 area 2
   * R3(config-router)#exit
10. Configure Loopback Interfaces
11. On Router R1:

* R1(config)#interface loopback 0
* R1(config-if)#ip address 172.16.1.252 255.255.0.0
* R1(config-if)#no shutdown
* R1(config-if)#exit

1. On Router R2:
   * R2(config)#interface loopback 0
   * R2(config-if)#ip address 172.16.1.253 255.255.0.0
   * R2(config-if)#no shutdown
   * R2(config-if)#exit
2. On Router R3:
   * R3(config)#interface loopback 0
   * R3(config-if)#ip address 172.16.1.254 255.255.0.0
   * R3(config-if)#no shutdown
   * R3(config-if)#exit
3. Configure Virtual Link
4. On Router R1:
   * R1(config)#router ospf 1
   * R1(config-router)#area 1 virtual-link 2.2.2.2
   * R1(config-router)#exit
5. On Router R2:
   * R2(config)#router ospf 1
   * R2(config-router)#area 1 virtual-link 1.1.1.1
   * R2(config-router)#exit
6. Verify Routing Table
7. On Router R3
8. R3#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area Gateway of last resort is not set

O 20.0.0.0/8 [110/128] via 30.0.0.1, 00:01:56, Serial1/0

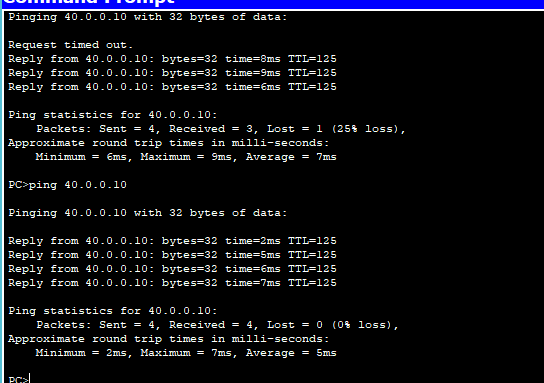
C 40.0.0.0/8 is directly connected, FastEthernet2/0

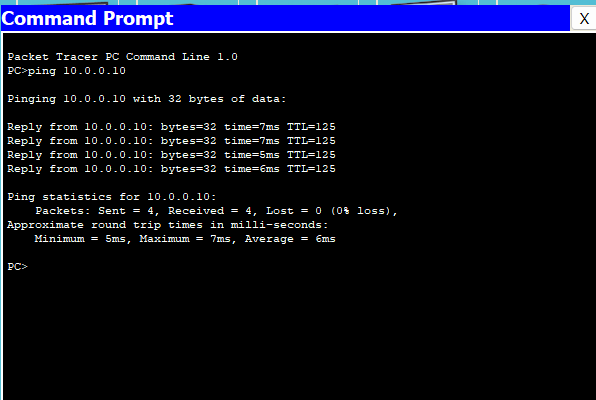
O 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:56, Serial1/0

C 30.0.0.0/8 is directly connected, Serial1/0

1. Check connectivity between host l 0.0.0. l 0 to 40.0.0.10

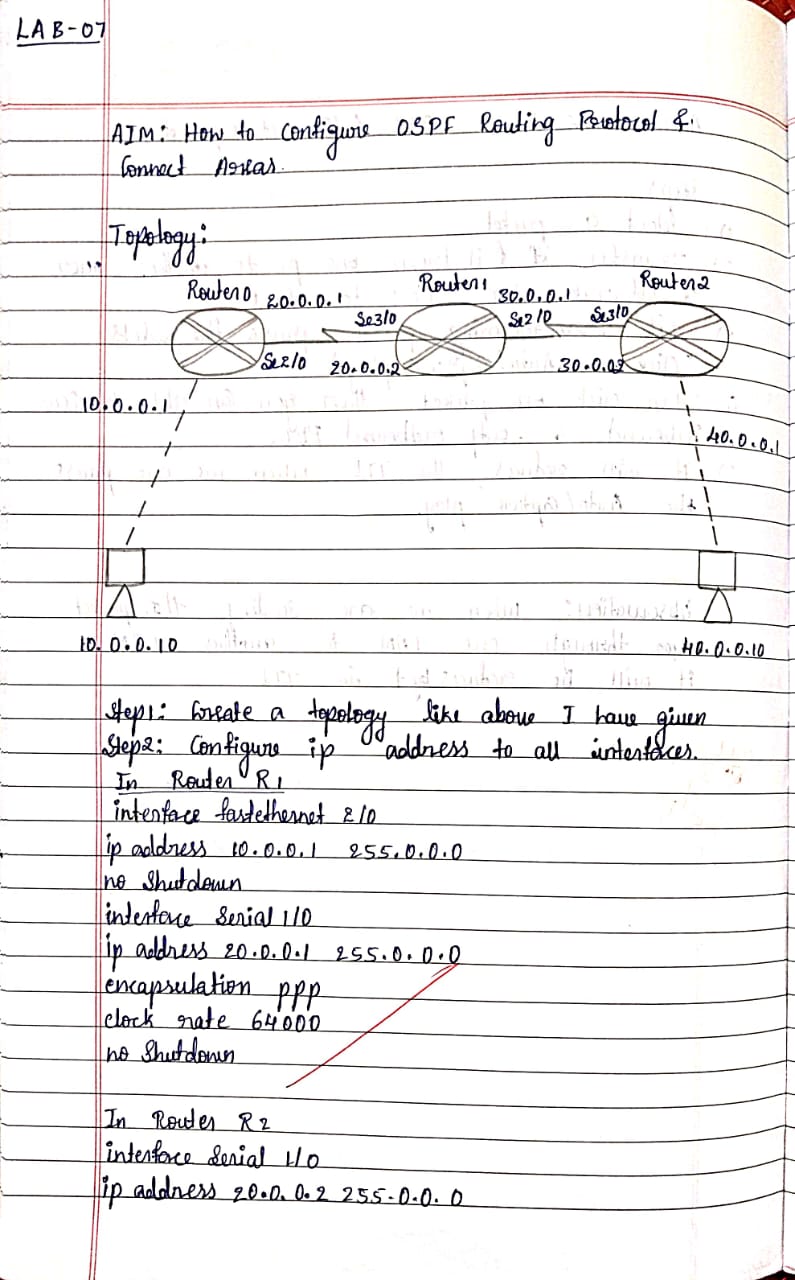
# Screen shots/ output :

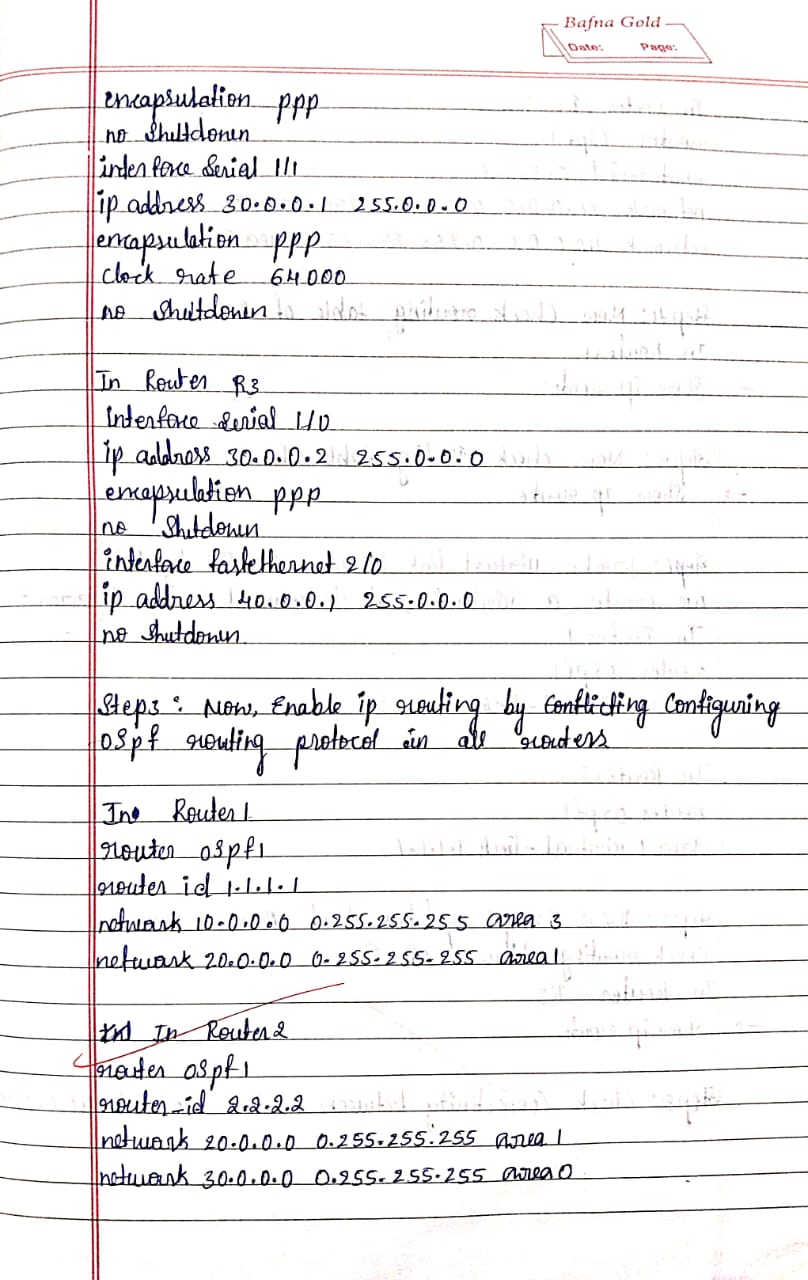
****

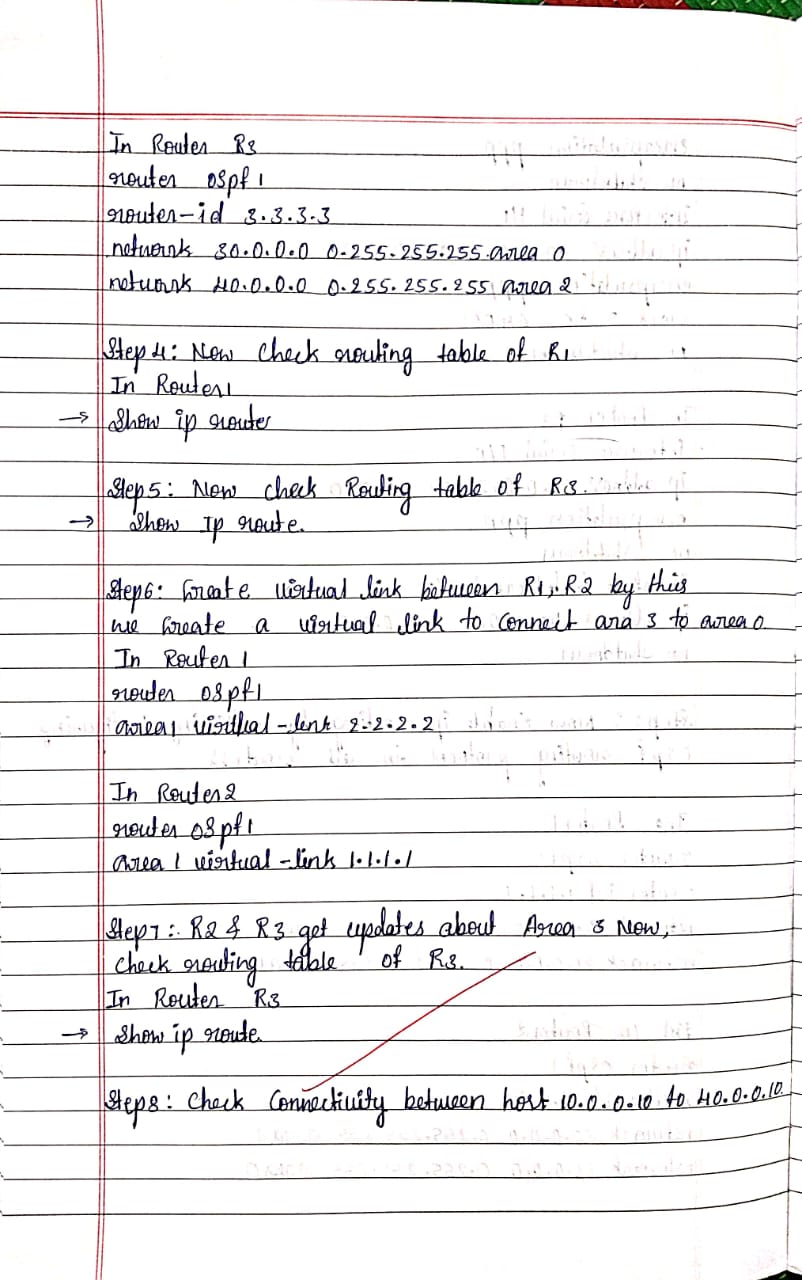


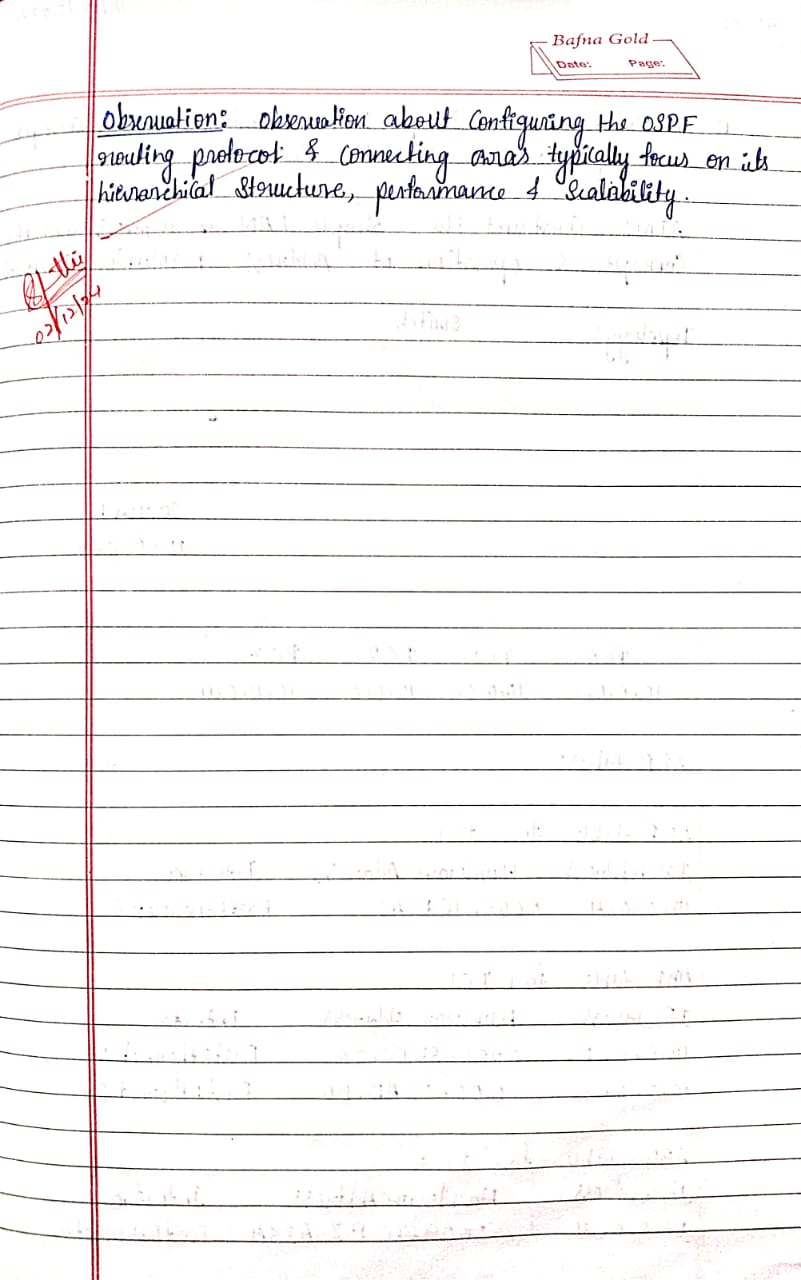
**Observations:**

* Check connectivity between host 10.0.0.10 to 40.0.0.10
* PC> ping 40.0.0.10 (In CP of PC0)
* Packets: sent = 4, received = 4, lost = 0 (0% loss)
* PC> ping 10.0.0.10 (In CP of PC4)
* Packets: sent = 4, received = 4, lost = 0 (0% loss)

****

****

****

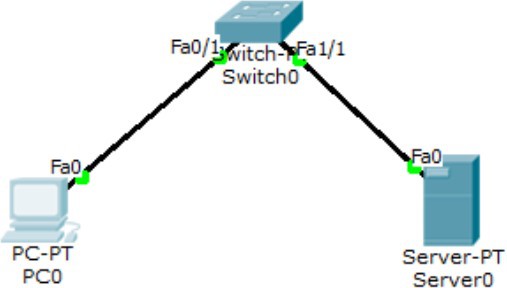
****

**Program 9**

# Aim of the program:

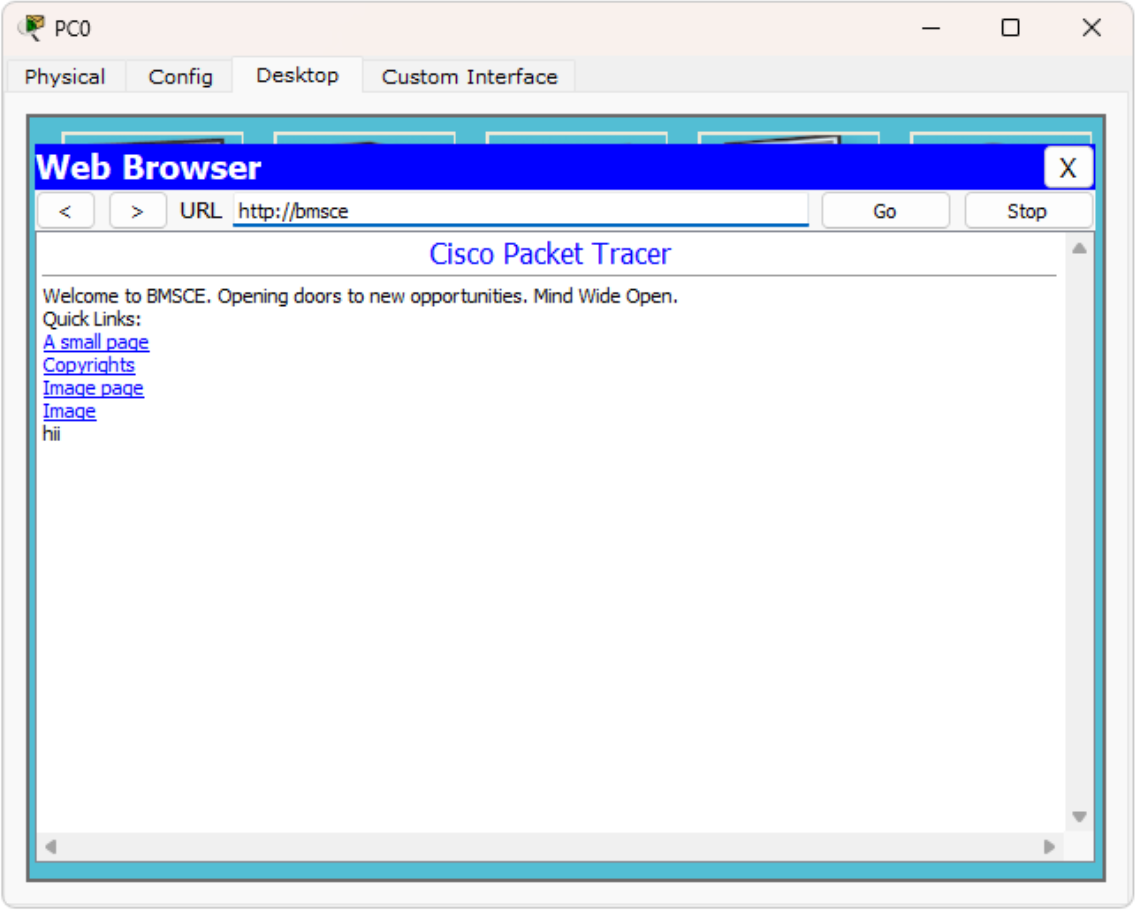
Configure Web Server, DNS within a LAN.

# Procedure along with the topology:

****

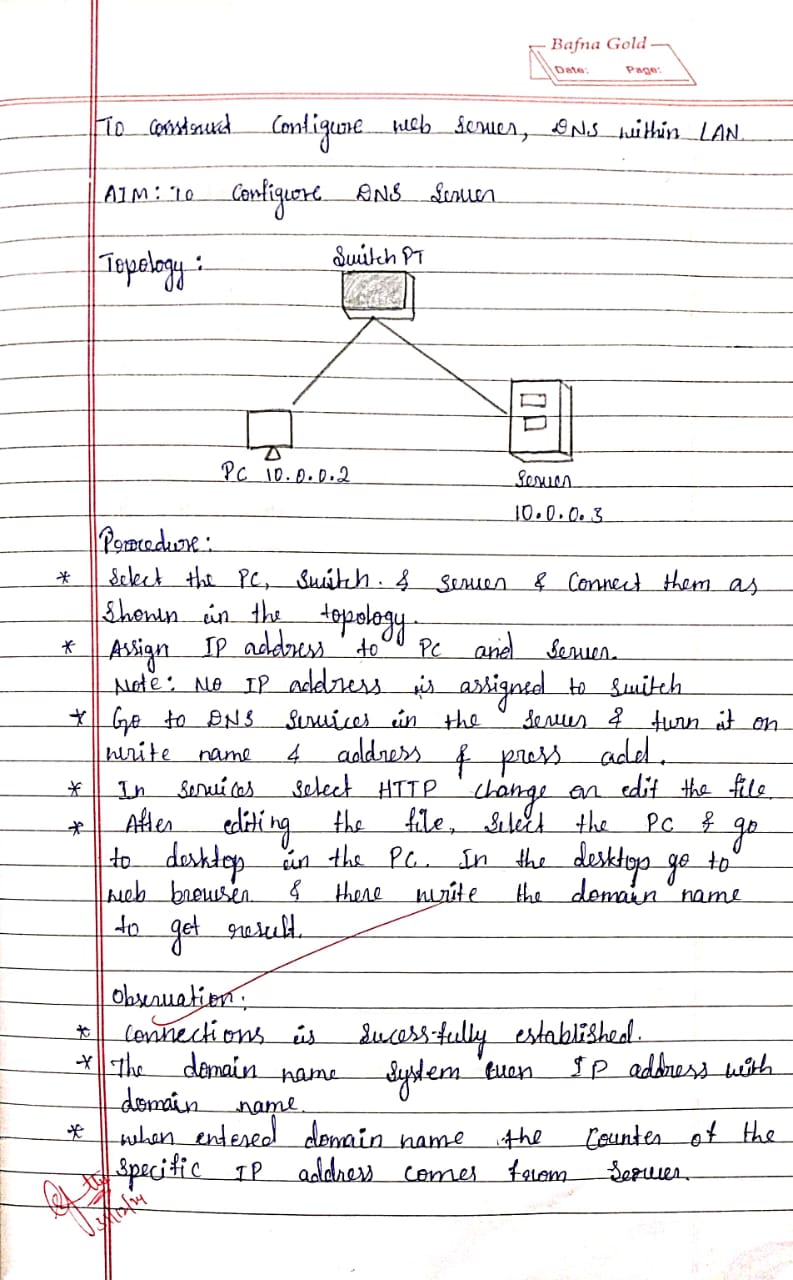
1. Create the above topology using PC, server, and switch.
2. Set the IP address of PC normally.
3. To set the IP of the server, go to config then select static method, set IP, and make sure port status is on.
4. Ping server from PC (ping is successful, 0% loss).
5. Now go to PC -> Desktop -> web server.
6. In the virtual browser type in the ip address of server and click.
7. Now web page is visible.

# Screen shots/ output :

****

**Observations:**

* + Successfully accessed the server's web page from PC (web server) by entering ip address.
  + Created simple LAN with PC, switch and server.

****

**Program 10**

# Aim of the program:

To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

# Procedure along with the topology:

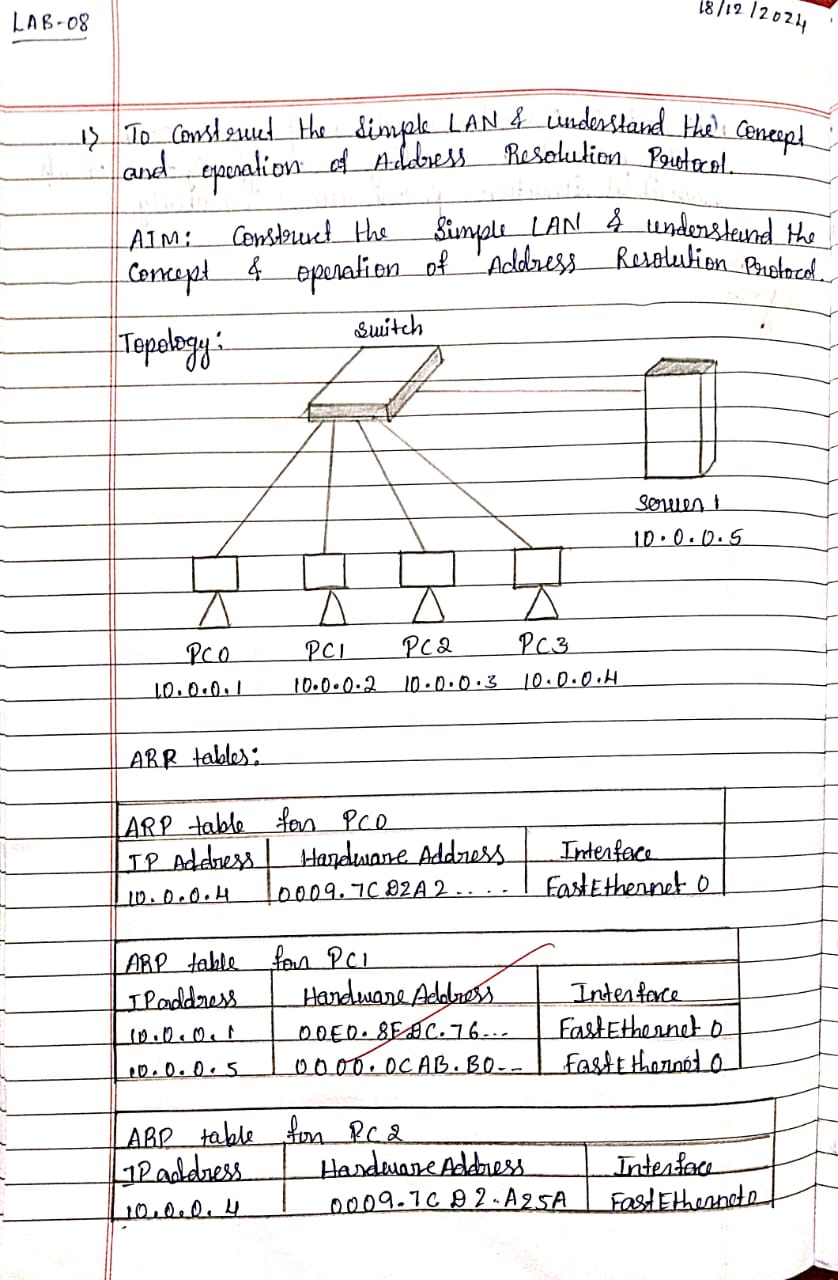
****

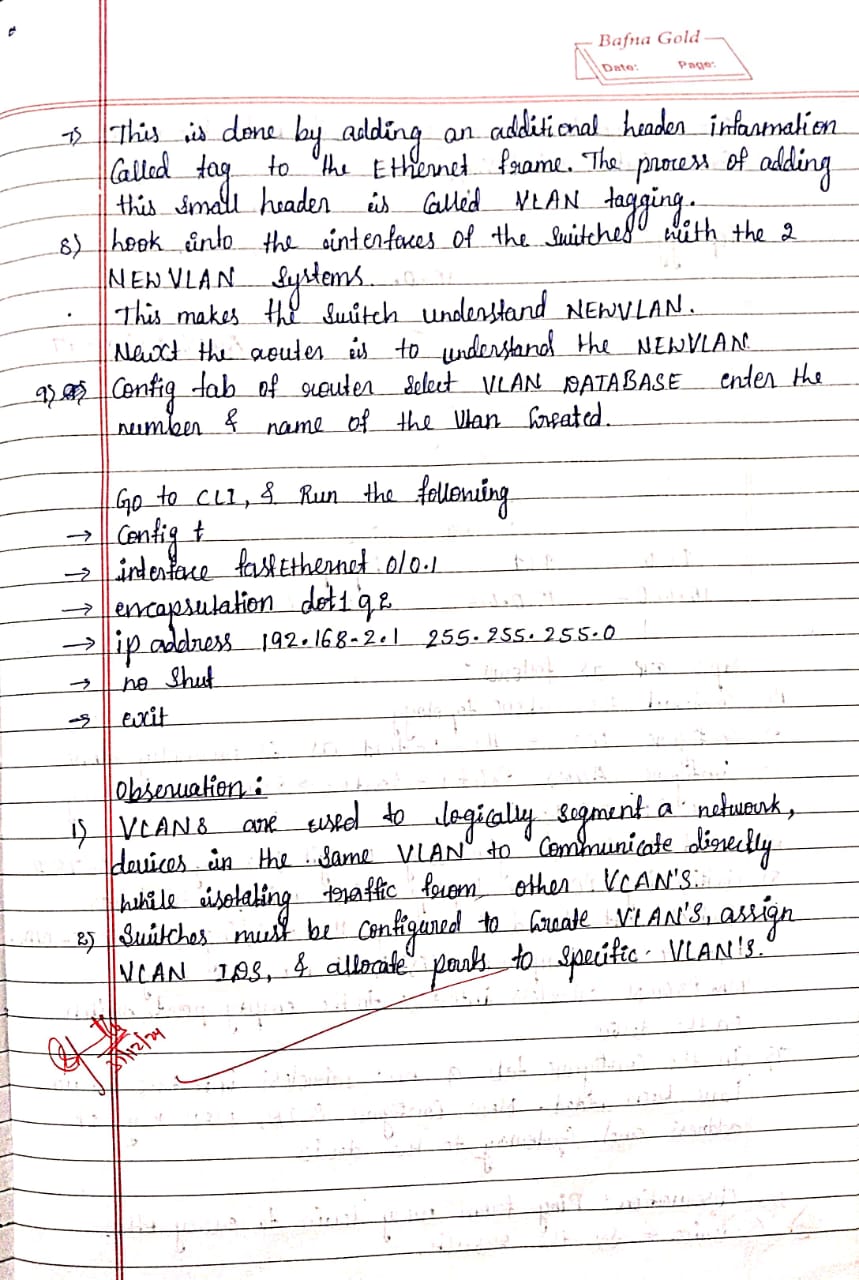
1. Create a topology of 4 PCs and a server and switch.
2. Assign IP to all.
3. Connect them through a switch.
4. Use the inspect tool to click on a PC to see the ARP table (command in CLI for same is arp -a).
5. Initially ARP table is empty (Arp in CLI of both, this command show mac address table can be given on every transaction to see how the switch learns from transaction and build the address table)
6. Use the capture in the simulation panel to go step by step so that changes in ARP can be clearly noted

# Screen shots/ output :

**Observations:**

Switch as well the nodes update the ARP table as and when new communication starts.

****

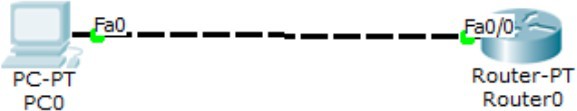
****

**Program 11**

# Aim of the program:

To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

# Procedure along with the topology:

****

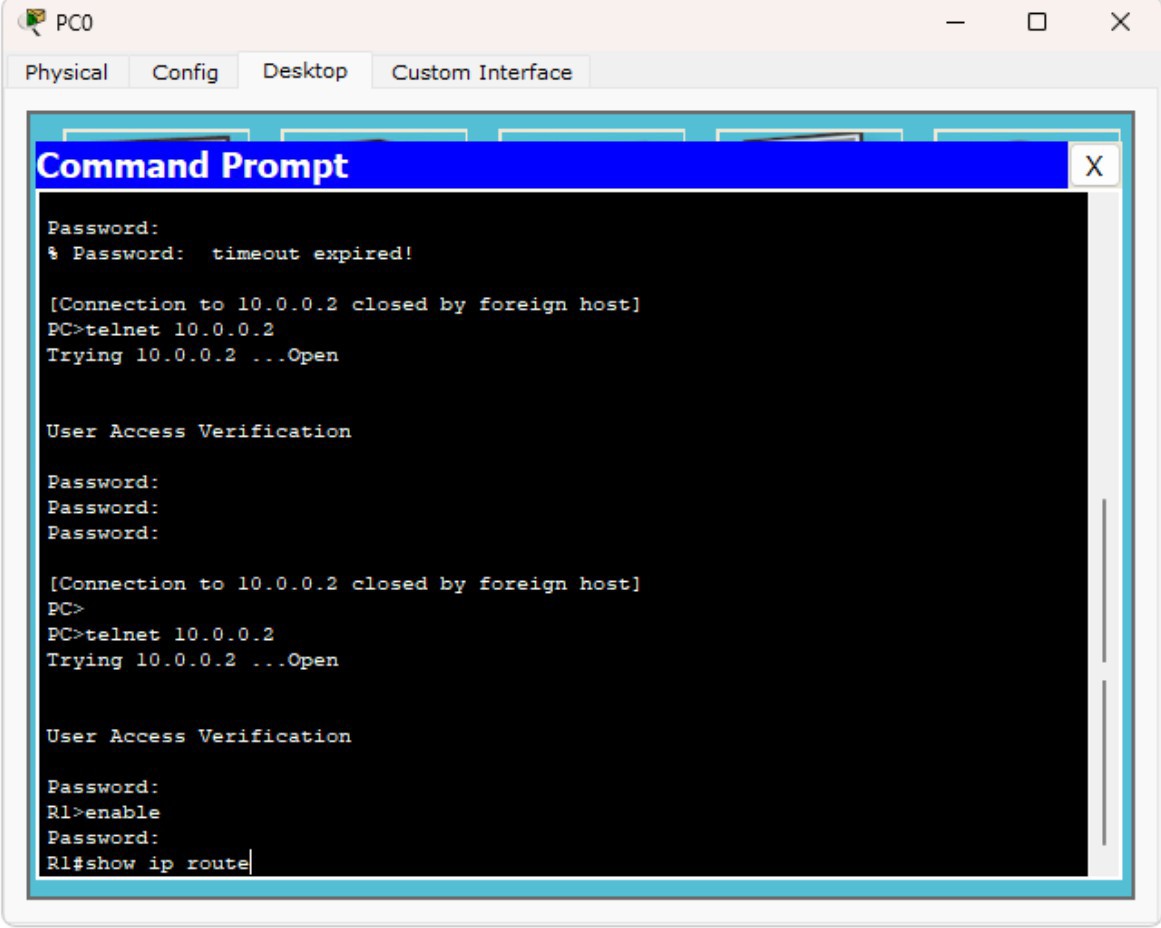
1. Make simple topology shown above.
2. Commands in router
   * enable # config terminal # hostname
   * R1 # enable secret Pl
   * # interface fastethernet 0/0
   * # ip address 10.0.0.1 255.0.0.0
   * # no shut
   * # line vty 0 5
   * # login
   * #password PO
   * #exit
   * >exit
   * wr --> to save changes
3. commands in pc

* ping 10.0.0.1
* ping results:- (0% loss)

1. (password for user access verification is PO) ,(password for enable is P1)
2. Accessing router CLI from PC

* user access verification
* password:
* r1> enable
* password:
* r1#show ip route

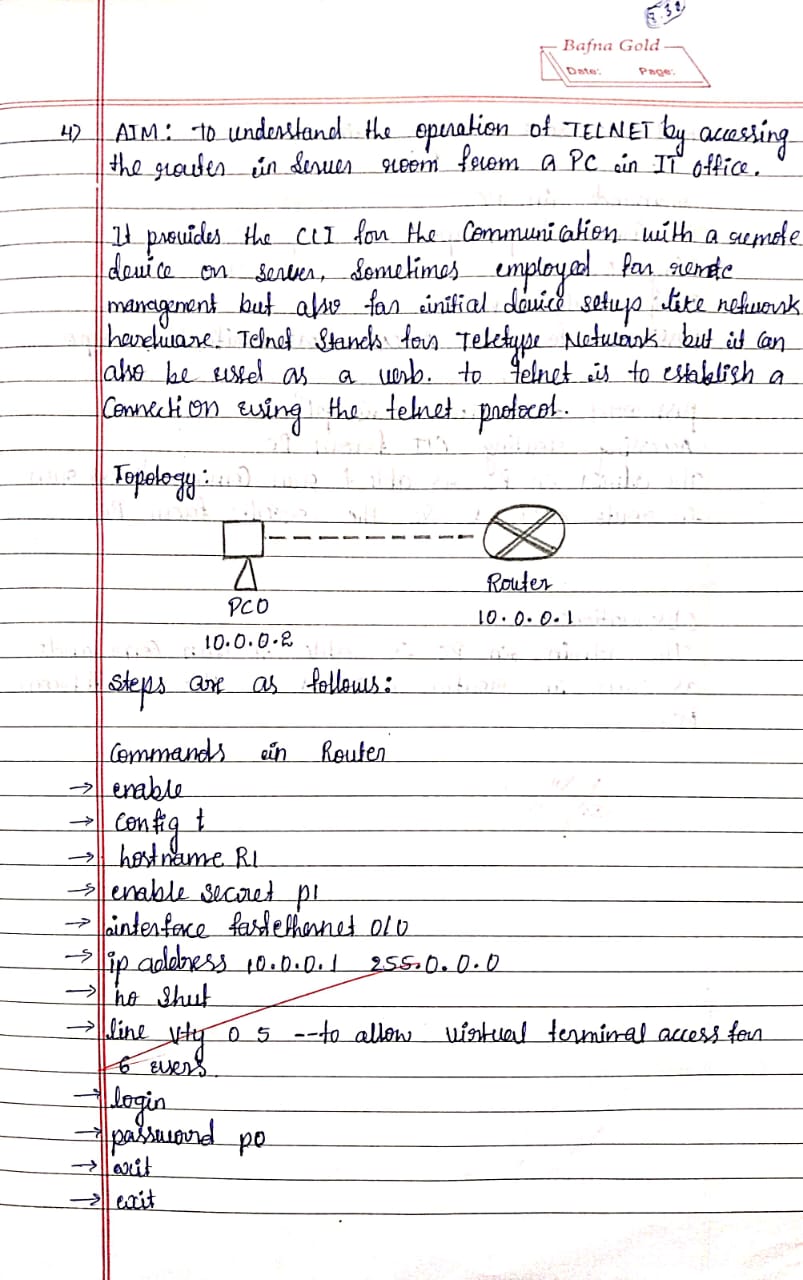
# Screen shots/ output :

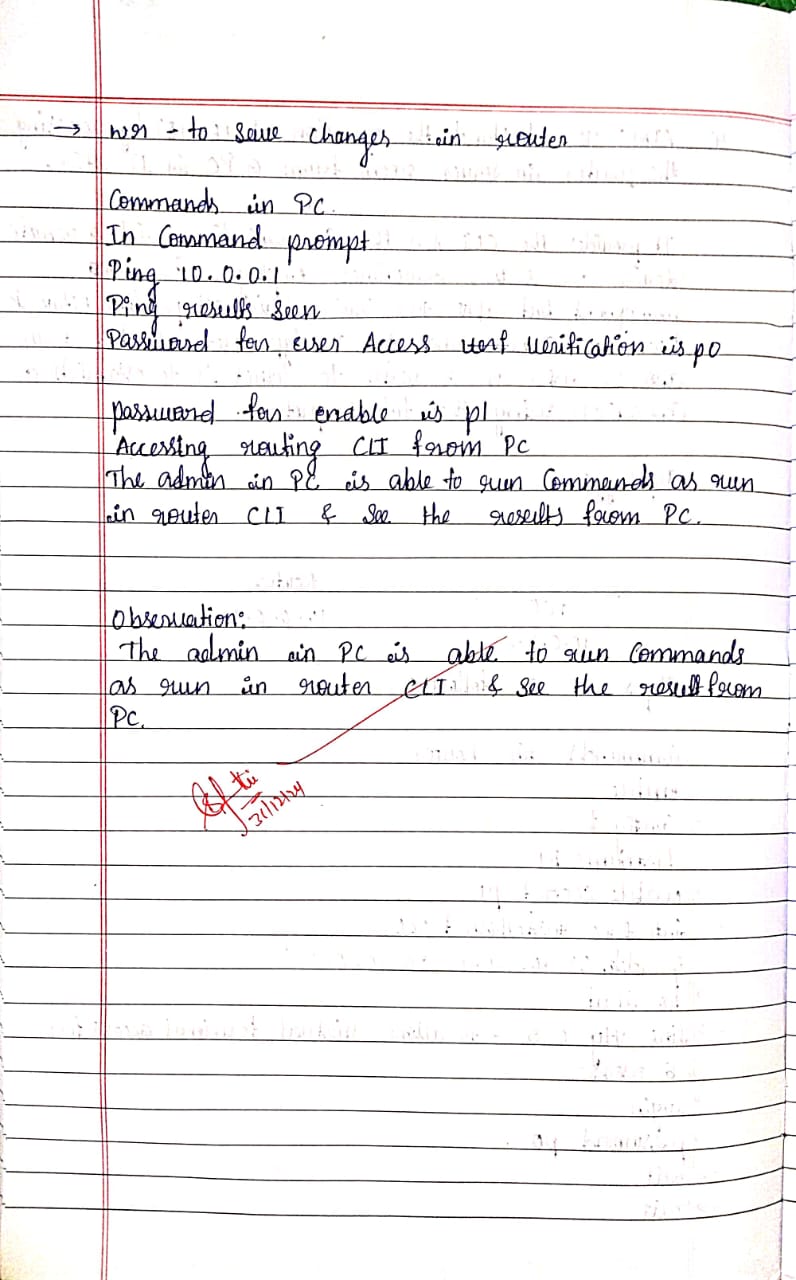
****

**Observations:**

The admin in PC is able to run commands as run in router CLI and see the result from PC.

**Observation Book:**

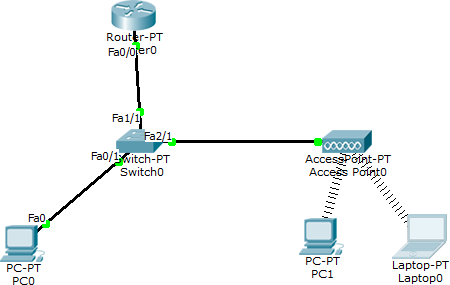
****

**Program 12**

# Aim of the program:

To construct a WLAN and make the nodes communicate wirelessly

# Procedure along with the topology:

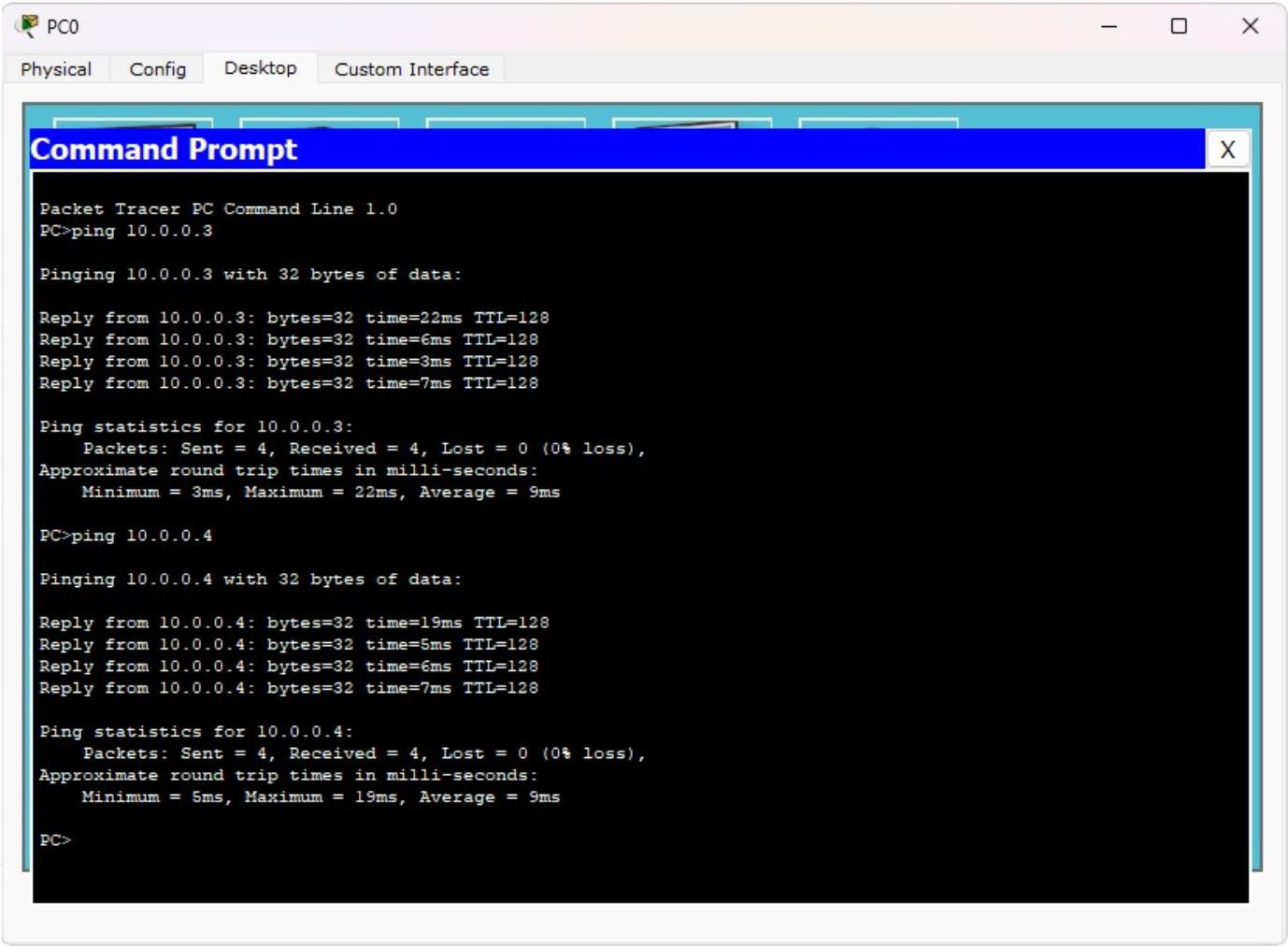
****

1. Construct the above topology.
2. Configure PC and Router as normally done.
3. Configure access point - Port1 -> SSID Name - any name (WLANhere)
4. Select WEP and give any 10 digit hex key (1234567890 here)
5. Configure PC1 and laptop with wireless standards
6. In PC1 Switch off the device, drag the existing PT-HOST-NM-1AM to the component listed in LHS.
7. Drag WMP300N wireless interface to the empty port. Switch on PC
8. In the config tab a new wireless interface would have been added. Now configure SSID,

WEP, WEP key, IP address and gateway (as normally done) to the device

1. Do similar in laptop.
2. Ping from every device to every other to check the result

# Screen shots/ output :

****

**Observations:**

* + Device could connect to WLAN as long as they are in the network range
  + Signal strength decreased with increase in distance
  + Ping is successful

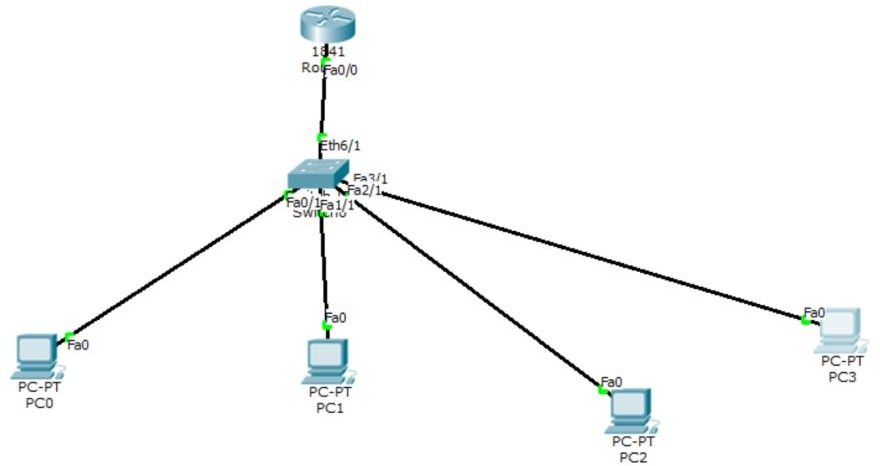
**Observation Book:**

**Program 13**

# Aim of the program:

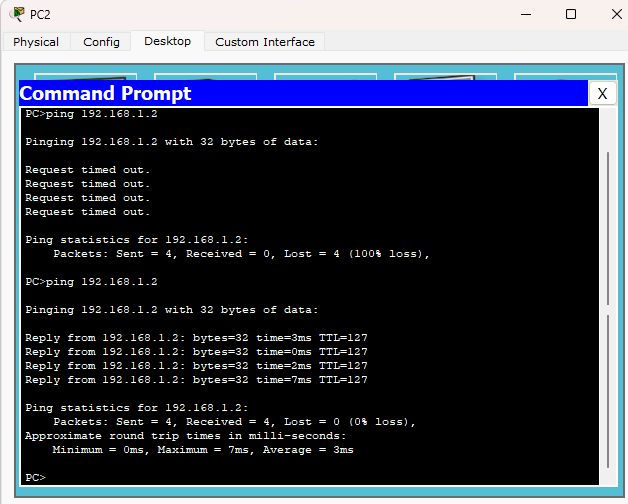
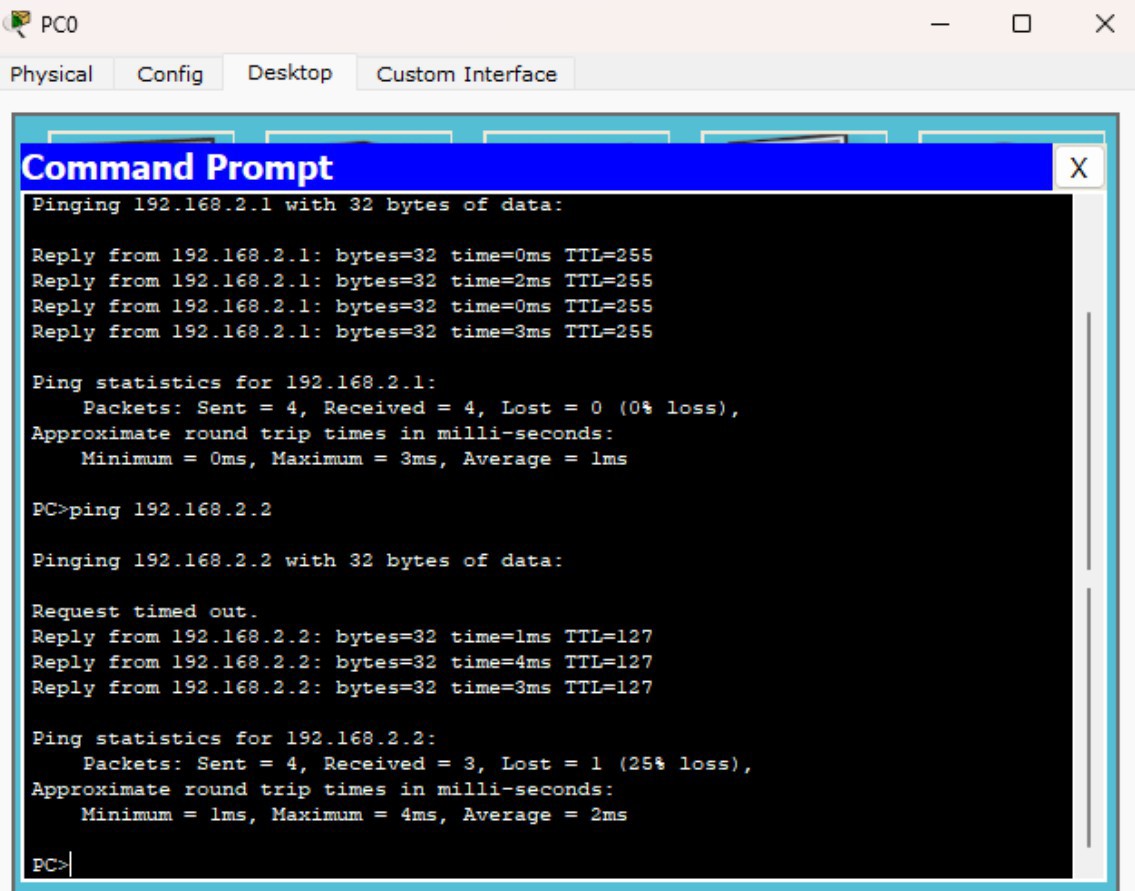
To construct a VLAN and make the PC’s communicate among a VLAN

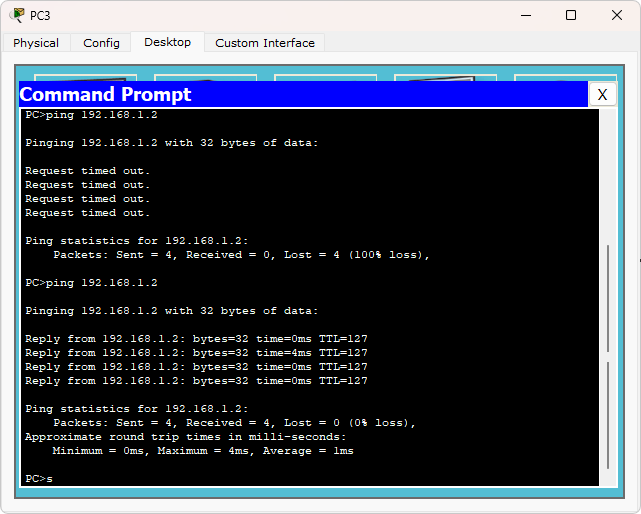
# Procedure along with the topology:

****

1. Create the topology seen above using router 1841
2. Assign IPs as shown in the topology
3. Go to the switch, choose VLAN database to configure the VLAN
4. Give VLAN number and VLAN name add it
5. Select the interface i.e. fastethernet0/1 (near the switch from router) and make it trunk. (VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk)
6. To make router understand VLAN,
   * Go to config tab of router, select VLAN database, enter the number and name of VLAN created.
   * Go to CLI
   * Router(vlan)#exit
   * Router#config t
   * Router(config)#interface fastethernet0/1
   * Router(config-subif)# #encapsulation dot1q 2
   * #ip address 192.168.21 255.255.255.0
   * #no shut
   * #exit
   * Router(config)#exit

# Screen shots/ output :

****



**Observations:**

* + Proper trunk configuration is established to make VLAN work properly
  + Ping from one VLAN to another works properly

# Observation Book:

# C:\Users\CHANDRAKALA\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\C900CED7451DA79502D29AA37EBB7B60\WhatsApp Image 2025-01-03 at 16.28.03_daf5e5b8.jpg

# C:\Users\CHANDRAKALA\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\1BC2029A8851AD344A8D503930DFD7F7\WhatsApp Image 2025-01-03 at 16.28.05_2aadf42b.jpg

# CYCLE – 2 PROGRAMS

**Program 14**

# Aim of the program

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

#include <iostream> #include <string.h> using namespace std;

int crc(char \*ip, char \*op, char \*poly, int mode)

{

strcpy(op, ip); if (mode) {

for (int i = 1; i < strlen(poly); i++) strcat(op, "0");

}

/\* Perform XOR on the msg with the selected polynomial \*/ for (int i = 0; i < strlen(ip); i++) {

if (op[i] == '1') {

for (int j = 0; j < strlen(poly); j++) { if (op[i + j] == poly[j])

op[i + j] = '0'; else

op[i + j] = '1';

}

}

}

/\* check for errors. return 0 if error detected \*/ for (int i = 0; i < strlen(op); i++)

if (op[i] == '1') return 0;

return 1;

}

int main()

{

char ip[50], op[50], recv[50];

/\* x 16 + x12 + x5 + 1 \*/

char poly[] = "10001000000100001";

cout << "Enter the input message in binary"<< endl; cin >> ip;

crc(ip, op, poly, 1);

cout << "The transmitted message is: " << ip << op + strlen(ip) << endl; cout << "Enter the received message in binary" << endl;

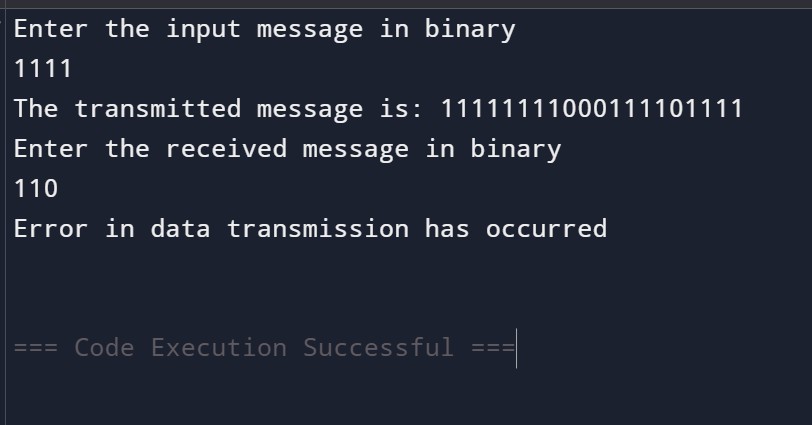
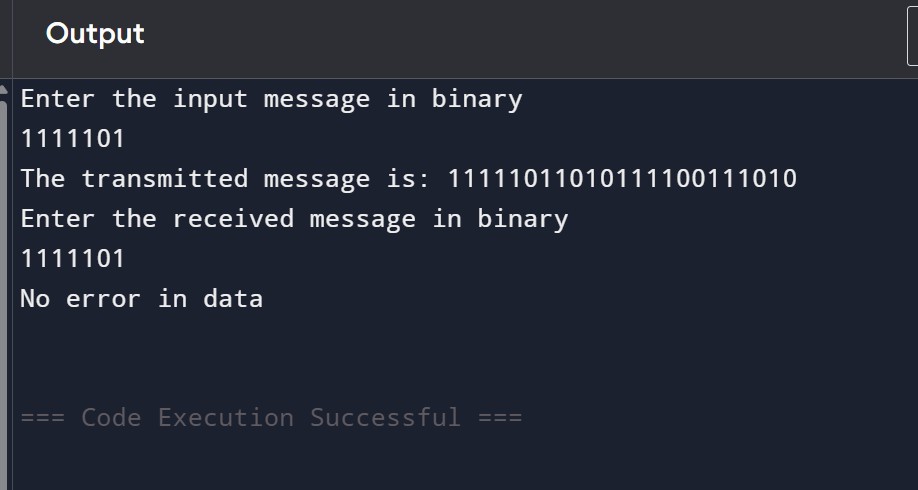
cin >> recv;

if (crc(recv, op, poly, 0))

cout << "No error in data" << endl; else

cout << "Error in data transmission has occurred" << endl; return 0;}

# Output

****

**Program 15**

# Aim of the program

Write a program for congestion control using Leaky bucket algorithm.

#include <iostream> #include <cstdlib> #include <unistd.h> using namespace std;

#define NOF\_PACKETS 10

// Function to generate random numbers within a range int generateRandom(int range) {

int rn = (rand() % range);

return rn == 0 ? 1 : rn; // Ensure non-zero values

}

int main() {

int packet\_sz[NOF\_PACKETS], i, clk, b\_size, o\_rate, p\_sz\_rm = 0, p\_time, op;

// Generate random packet sizes

for (i = 0; i < NOF\_PACKETS; ++i)

packet\_sz[i] = generateRandom(6) \* 10;

// Display packet sizes

for (i = 0; i < NOF\_PACKETS; ++i)

cout << "\nPacket[" << i << "]: " << packet\_sz[i] << " bytes";

// Input: Output rate and bucket size cout << "\nEnter the Output rate: "; cin >> o\_rate;

cout << "Enter the Bucket Size: "; cin >> b\_size;

// Process each packet

for (i = 0; i < NOF\_PACKETS; ++i) {

// Check if the packet size exceeds bucket size if ((packet\_sz[i] + p\_sz\_rm) > b\_size) {

if (packet\_sz[i] > b\_size) {

// Packet size exceeds bucket capacity

cout << "\n\nIncoming packet size (" << packet\_sz[i] << " bytes) exceeds bucket capacity ("

<< b\_size << " bytes) - PACKET REJECTED!";

} else {

// Bucket overflow

cout << "\n\nBucket capacity exceeded - PACKETS REJECTED!";

}

} else {

// Accept packet into the bucket p\_sz\_rm += packet\_sz[i];

cout << "\n\nIncoming Packet size: " << packet\_sz[i]; cout << "\nBytes remaining to transmit: " << p\_sz\_rm;

// Simulate transmission

p\_time = generateRandom(4) \* 10;

cout << "\nTime allocated for transmission: " << p\_time << " units";

for (clk = 10; clk <= p\_time; clk += 10) { sleep(1); // Simulate time delay

if (p\_sz\_rm > 0) {

if (p\_sz\_rm <= o\_rate) { op = p\_sz\_rm; p\_sz\_rm = 0;

} else {

op = o\_rate; p\_sz\_rm -= o\_rate;

}

cout << "\nPacket of size " << op << " transmitted."; cout << " Bytes remaining to transmit: " << p\_sz\_rm;

} else {

cout << "\nTime left for transmission: " << (p\_time - clk) << " units";

cout << "\nNo packets to transmit!";

}

}

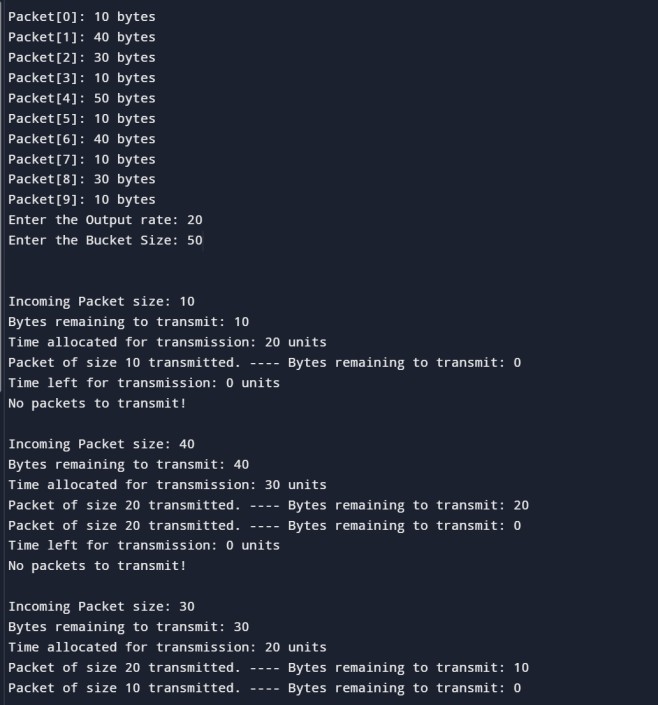
}

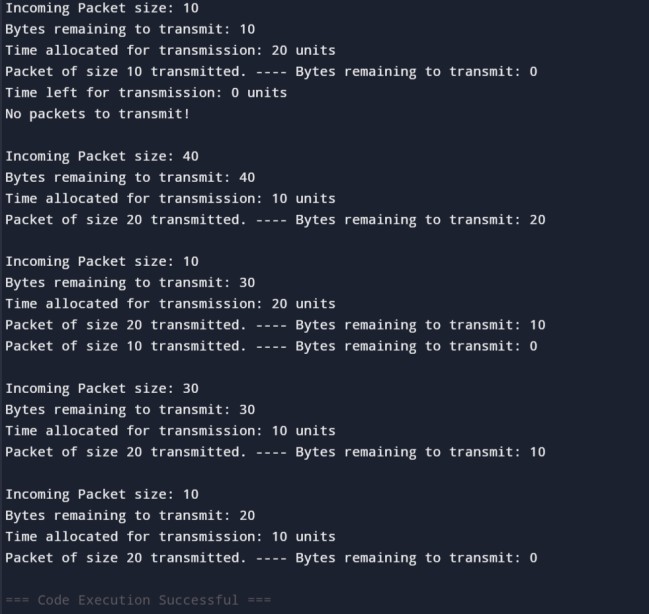
}

return 0;

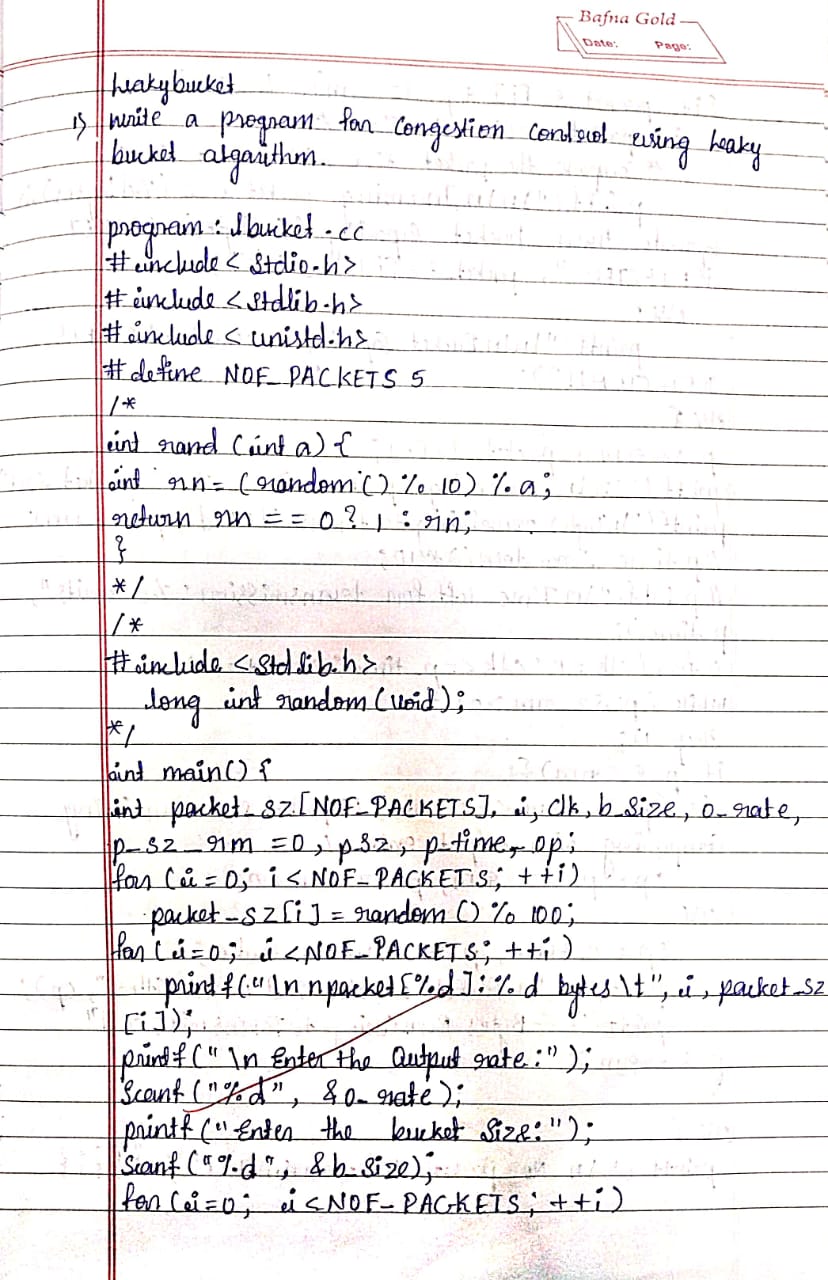
}

# Output

****

****

**Observation Book:**

****

****

**Program 16**

# Aim of the program

Using TCP/IP 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.

Client Side

#include <unistd.h> int main()

{

int soc, n;

char buffer[1024], fname[50]; struct sockaddr\_in addr;

/\* socket creates an endpoint for communication and returns a file descriptor \*/ soc = socket(PF\_INET, SOCK\_STREAM, 0);

/\*

* sockaddr\_in is used for ip manipulation
* we define the port and IP for the connection.

\*/

addr.sin\_family = AF\_INET; addr.sin\_port = htons(7891);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

/\* keep trying to esatablish connection with server \*/ while(connect(soc, (struct sockaddr \*) &addr, sizeof(addr))) ; printf("\nClient is connected to Server");

printf("\nEnter file name: "); scanf("%s", fname);

/\* send the filename to the server \*/ send(soc, fname, sizeof(fname), 0); printf("\nRecieved response\n");

/\* keep printing any data received from the server \*/ while ((n = recv(soc, buffer, sizeof(buffer), 0)) > 0) printf("%s", buffer);

return 0;

}

## Server Side

#include <stdio.h> #include <arpa/inet.h> #include <fcntl.h> #include <unistd.h> int main()

{

int welcome, new\_soc, fd, n; char buffer[1024], fname[50]; struct sockaddr\_in addr;

welcome = socket(PF\_INET, SOCK\_STREAM, 0);

addr.sin\_family = AF\_INET; addr.sin\_port = htons(7891);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1"); bind(welcome, (struct sockaddr \*) &addr, sizeof(addr)); printf("\nServer is Online");

/\* listen for connections from the socket \*/ listen(welcome, 5);

/\* accept a connection, we get a file descriptor \*/ new\_soc = accept(welcome, NULL, NULL);

/\* receive the filename \*/ recv(new\_soc, fname, 50, 0);

printf("\nRequesting for file: %s\n", fname);

/\* open the file and send its contents \*/ fd = open(fname, O\_RDONLY);

if (fd < 0)

send(new\_soc, "\nFile not found\n", 15, 0); else

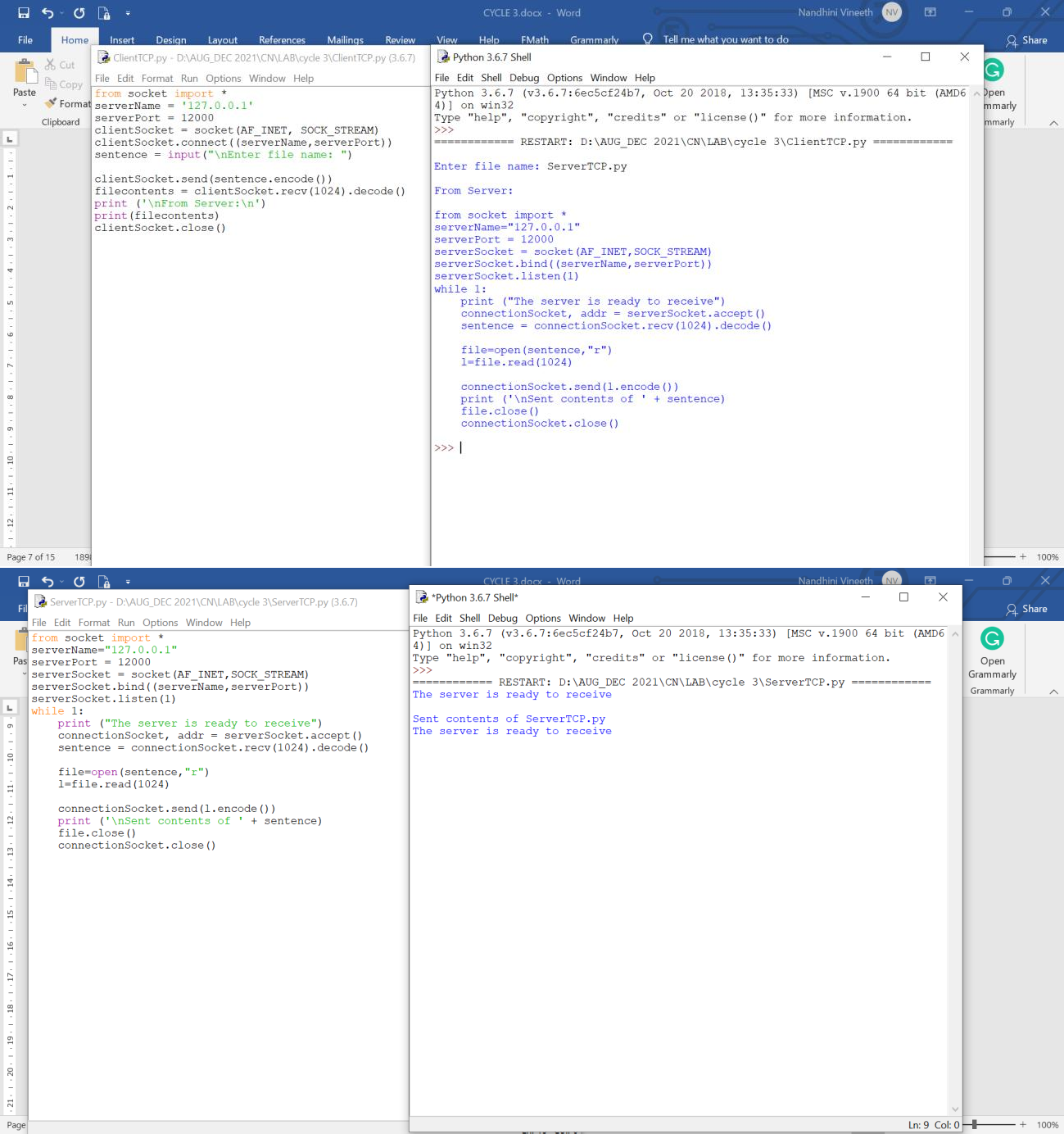
while ((n = read(fd, buffer, sizeof(buffer))) > 0) send(new\_soc, buffer, n, 0);

printf("\nRequest sent\n"); close(fd);

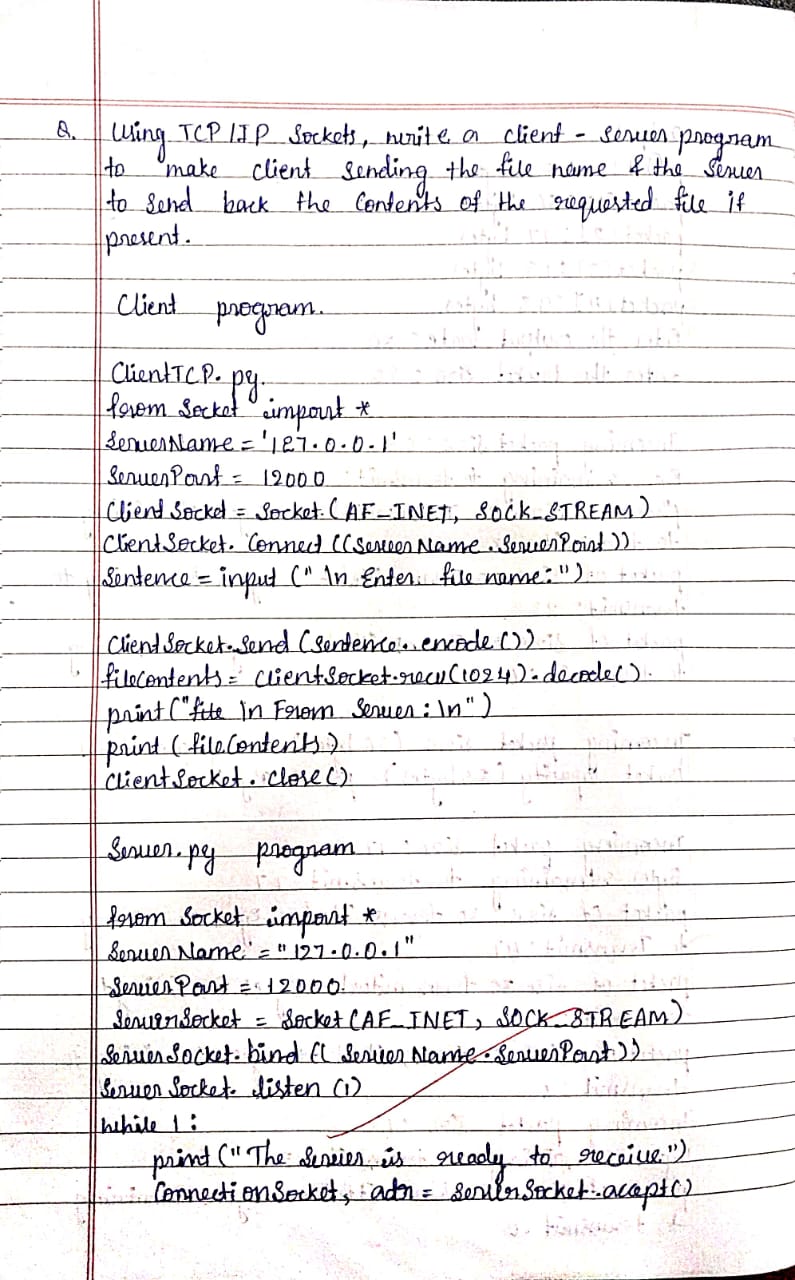
return 0;

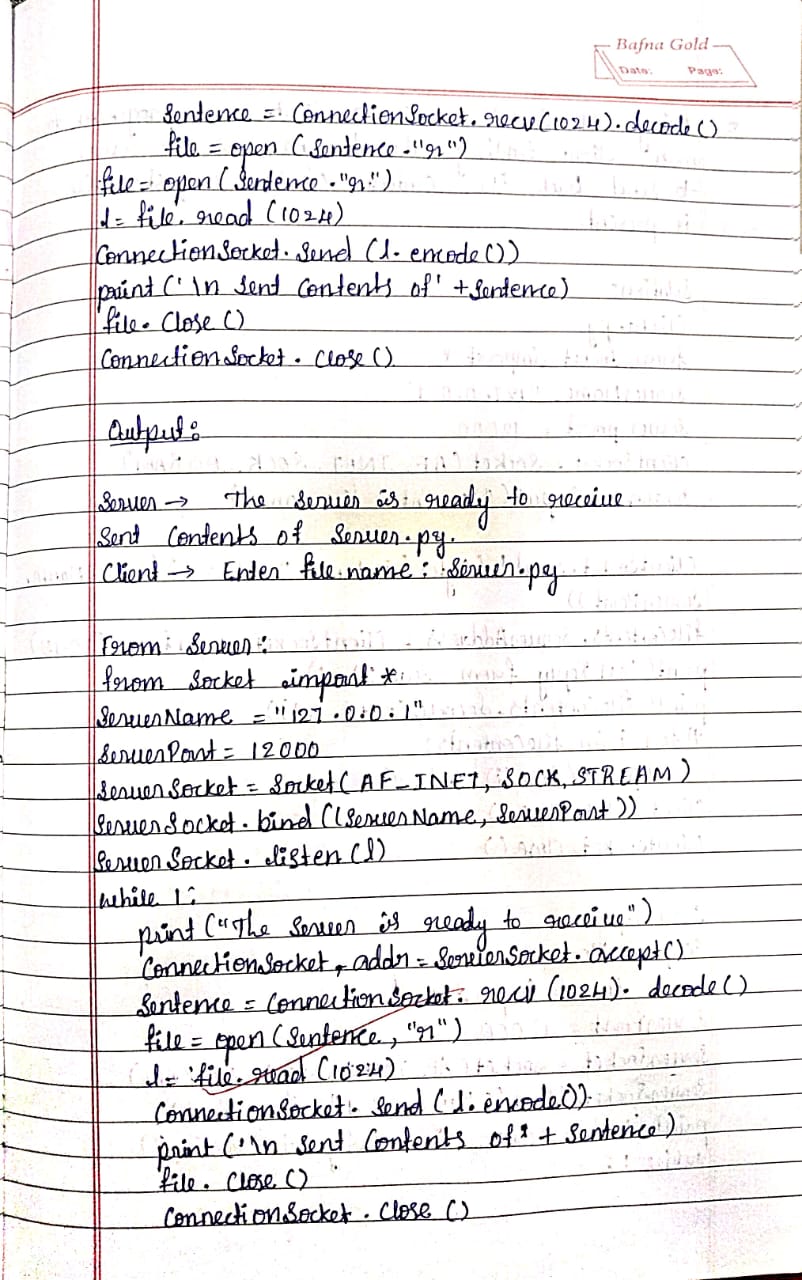
}

**OUTPUT:**



**Observation Book:**

****



**Program 17**

# Aim of the program

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.

## Server program

#include <stdio.h> #include <strings.h> #include <sys/types.h> #include <arpa/inet.h> #include <sys/socket.h> #include<netinet/in.h> #define PORT 5000

#define MAXLINE 1000

// Driver code int main()

{

char buffer[100];

char \*message = "Hello Client"; int listenfd, len;

struct sockaddr\_in servaddr, cliaddr; bzero(&servaddr, sizeof(servaddr));

// Create a UDP Socket

listenfd = socket(AF\_INET, SOCK\_DGRAM, 0); servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY); servaddr.sin\_port = htons(PORT); servaddr.sin\_family = AF\_INET;

// bind server address to socket descriptor

bind(listenfd, (struct sockaddr\*)&servaddr, sizeof(servaddr));

//receive the datagram len = sizeof(cliaddr);

int n = recvfrom(listenfd, buffer, sizeof(buffer),

0, (struct sockaddr\*)&cliaddr,&len); //receive message from server buffer[n] = '\0';

puts(buffer);

// send the response

sendto(listenfd, message, MAXLINE, 0, (struct sockaddr\*)&cliaddr, sizeof(cliaddr));

}

## client driver program

#include <stdio.h> #include <strings.h> #include <sys/types.h> #include <arpa/inet.h>

#include <sys/socket.h> #include<netinet/in.h> #include<unistd.h> #include<stdlib.h> #define PORT 5000

#define MAXLINE 1000

// Driver code int main()

{

char buffer[100];

char \*message = "Hello Server"; int sockfd, n;

struct sockaddr\_in servaddr;

// clear servaddr

bzero(&servaddr, sizeof(servaddr)); servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1"); servaddr.sin\_port = htons(PORT); servaddr.sin\_family = AF\_INET;

// create datagram socket

sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);

// connect to server

if(connect(sockfd, (struct sockaddr \*)&servaddr, sizeof(servaddr)) < 0)

{

printf("\n Error : Connect Failed \n"); exit(0);

}

// request to send datagram

// no need to specify server address in sendto

// connect stores the peers IP and port

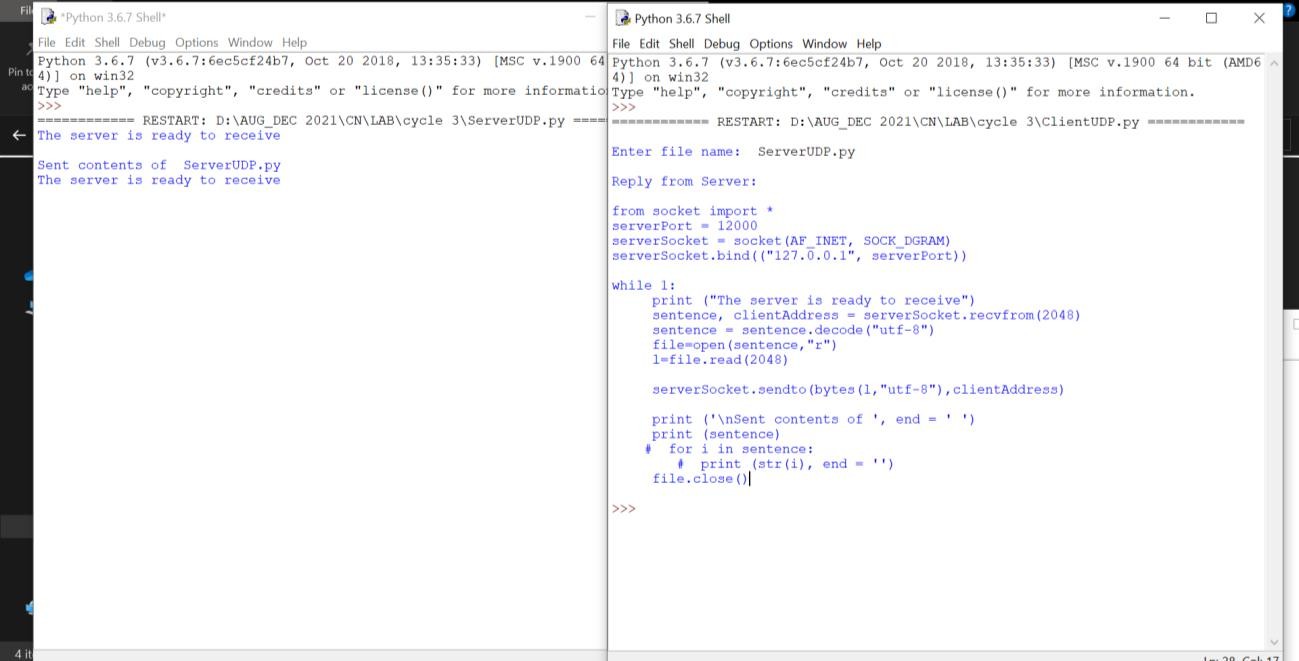
sendto(sockfd, message, MAXLINE, 0, (struct sockaddr\*)NULL, sizeof(servaddr));

// waiting for response

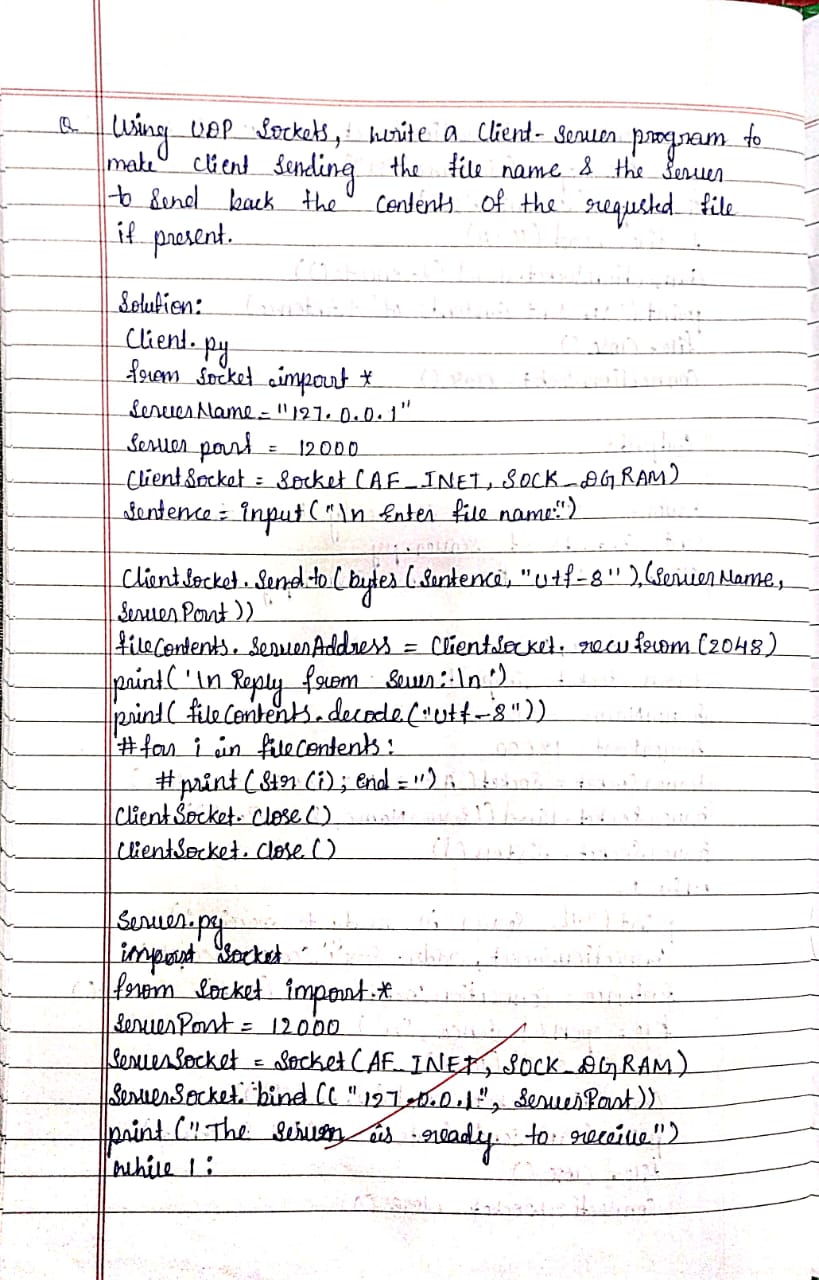
recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct sockaddr\*)NULL, NULL); puts(buffer);

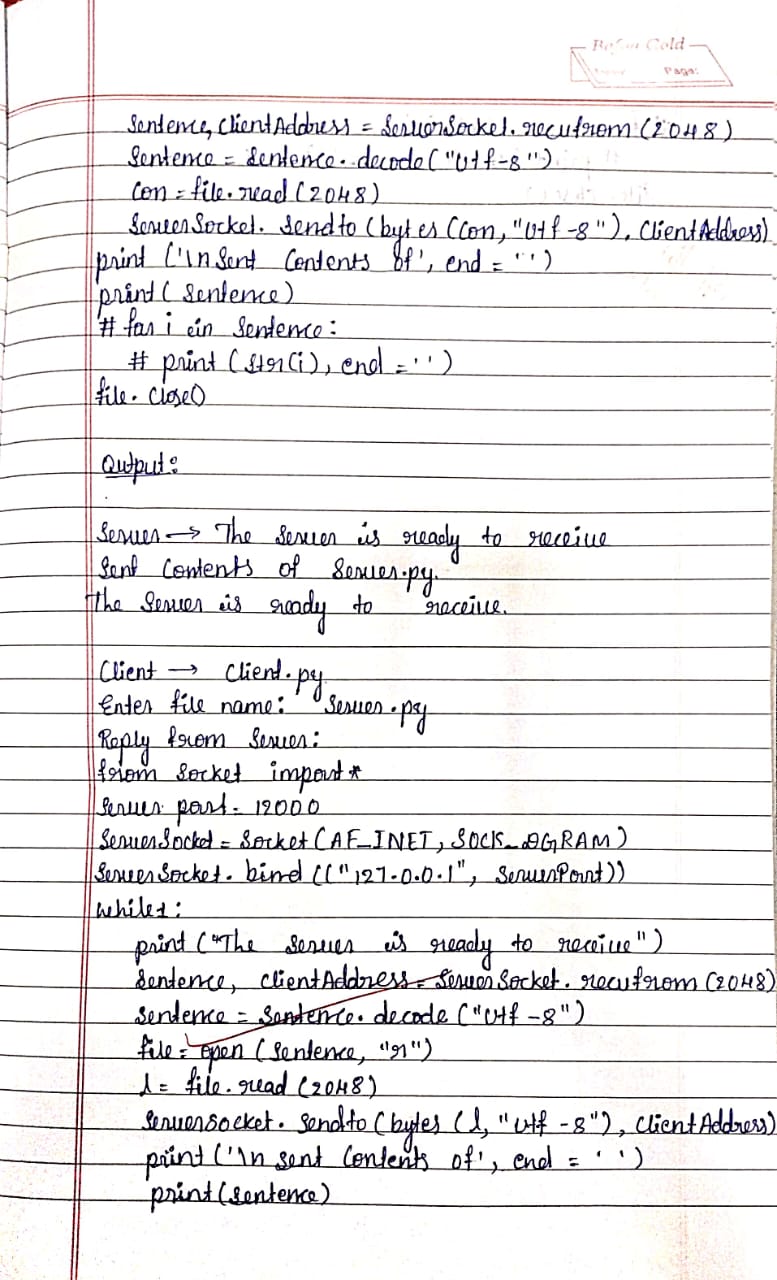
// close the descriptor close(sockfd);

**OUTPUT:**



**Observation Book:**



****