PRACTICAL NO. 1

Q. Demonstrate the Static Routing in Packet tracer.

Aim: To write code to demonstrate static routing in packet tracer.

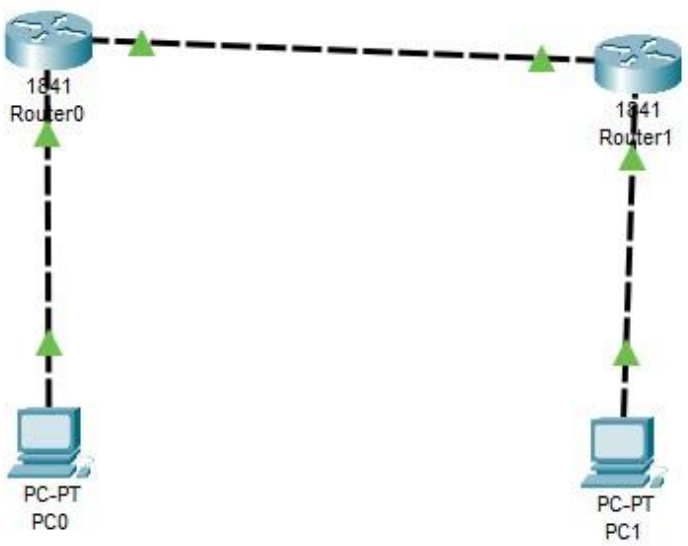
Steps: 1. In packet tracer place 2 PCs and 2 Routers.

2. Connect Router0 with PC0 via fastethernet0/0 using cross-copper cord and Router1 with PC1 via fastethernet0/0 using cross-copper cord.

3. Connect both router via fastethernet 0/1 using cross-copper cord.

4. Set the IP address and default gatway of both the pcs in ip configuration of both the pcs respectively.

Topology:



Code:

Router0:

Router>enable

Router#configure terminal

Router(config)#interface FastEthernet0/0

Router(config-if)#ip address 10.0.0.1 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface FastEthernet0/1

Router(config-if)#ip address 20.0.0.1 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2

Router(config)#exit

Router 1:

Router>enable

Router#configure terminal

Router#configure terminal

Router(config)#interface FastEthernet0/0

Router(config-if)#ip address 20.0.0.2 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface FastEthernet0/1

Router(config-if)#ip address 40.0.0.1 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router(config)#exit.

Verification: In Both PCs Command Prompt

PC1:

C:\ipconfig

C:\ping 10.0.0.1

C:\ping 20.0.0.1

C:\ping 20.0.0.2

C:\ping 40.0.0.1

C:\ping 40.0.0.2

PC2:

C:\ipconfig

C:\ping 40.0.0.1

C:\ping 20.0.0.2

C:\ping 20.0.0.1

C:\ping 10.0.0.1

C:\ping 10.0.0.2

Output:

Pc0:

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=126
Reply from 10.0.0.2: bytes=32 time<1ms TTL=126
Reply from 10.0.0.2: bytes=32 time<1ms TTL=126
Reply from 10.0.0.2: bytes=32 time<1ms TTL=126

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Pc1:

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=1ms TTL=126
Reply from 40.0.0.2: bytes=32 time<1ms TTL=126
Reply from 40.0.0.2: bytes=32 time<1ms TTL=126
Reply from 40.0.0.2: bytes=32 time<1ms TTL=126

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

Conclusion: The above network topology has been executed successfully.

PRACTICAL NO.2

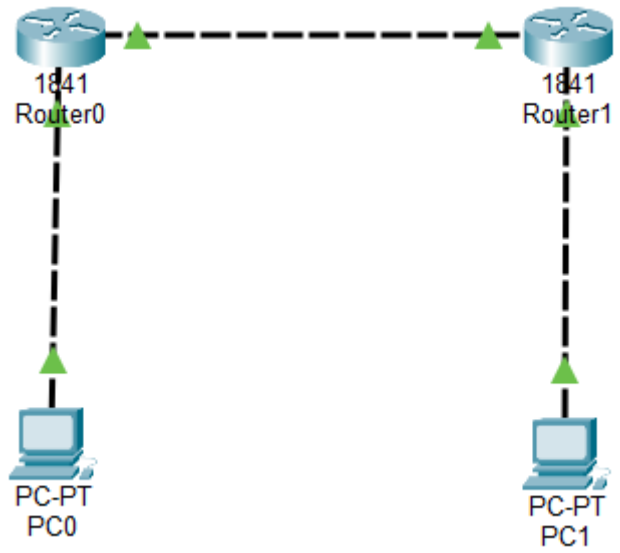
Q. Demonstrate OSPF dynamic routing protocol

Aim: To configure and demonstrate OSPF dynamic routing protocol between two networks using two routers in Packet Tracer.

Steps :

1. In packet tracer place 2 PCs and 2 Routers.
2. Connect Router0 with PC0 via FastEthernet0/0 using cross-copper cord and Router1 with PC1 via FastEthernet0/0 using cross-copper cord.
3. Connect both routers via FastEthernet0/1 using cross-copper cord.
4. Set the IP address and default gateway of both the PCs in IP configuration of both PCs respectively.
5. Configure OSPF routing protocol on both routers.

Topology :



Code :

Router0 configuration :

Router>enable

```
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface FastEthernet0/0
Router0(config-if)#ip address 192.168.1.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#interface FastEthernet0/1
Router0(config-if)#ip address 10.0.0.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#router ospf 1
Router0(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router0(config-router)#network 10.0.0.0 0.0.0.255 area 0
Router0(config-router)#exit
Router0(config)#exit
Router0#copy running-config startup-config
```

Router1 configuration :

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface FastEthernet0/1
Router1(config-if)#ip address 10.0.0.2 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
```

Router1(config)#interface FastEthernet0/0

Router1(config-if)#ip address 172.16.1.1 255.255.255.0

Router1(config-if)#no shutdown

Router1(config-if)#exit

Router1(config)#router ospf 1

Router1(config-router)#network 10.0.0.0 0.0.0.255 area 0

Router1(config-router)#network 172.16.1.0 0.0.0.255 area 0

Router1(config-router)#exit

Router1(config)#exit

Router1#copy running-config startup-config

PC0 configuration :

- IP Address: 192.168.1.2
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1
-

PC1 Configuration :

- IP Address: 172.16.1.2
- Subnet Mask: 255.255.255.0
- Default Gateway: 172.16.1.1
-

PC0 Command Prompt :

C:\>ipconfig

C:\>ping 192.168.1.1

C:\>ping 10.0.0.1

C:\>ping 10.0.0.2

C:\>ping 172.16.1.1

C:\>ping 172.16.1.2

PC1 Command Prompt :

C:\>ipconfig

C:\>ping 172.16.1.1

C:\>ping 10.0.0.2

C:\>ping 10.0.0.1

C:\>ping 192.168.1.1

C:\>ping 192.168.1.2

Router Verification Commands:

Router0#show ip route

Router0#show ip ospf neighbour

Router0#show ip ospf database

Router1#show ip route

Router1#show ip ospf neighbour

OUTPUT:

PC0 OUTPUT :

```
C:\>PING 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Request timed out.
Reply from 172.16.1.2: bytes=32 time<1ms TTL=126
Reply from 172.16.1.2: bytes=32 time<1ms TTL=126
Reply from 172.16.1.2: bytes=32 time<1ms TTL=126

Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC1 OUTPUT :

```
C:\>PING 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=126
Reply from 192.168.1.2: bytes=32 time<1ms TTL=126
Reply from 192.168.1.2: bytes=32 time<1ms TTL=126
Reply from 192.168.1.2: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

OSPF Neighbour Output:

```
Router#show ip ospf neighbor


Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.1.1        1   FULL/BDR        00:00:34    10.0.0.2        FastEthernet0/1
```

IP Route Output:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet0/1
O    172.16.0.0/16 [110/2] via 10.0.0.2, 00:07:45, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

Conclusion : The above network topology has been executed successfully and OSPF dynamic routing has been demonstrated.

PRACTICAL NO. 3

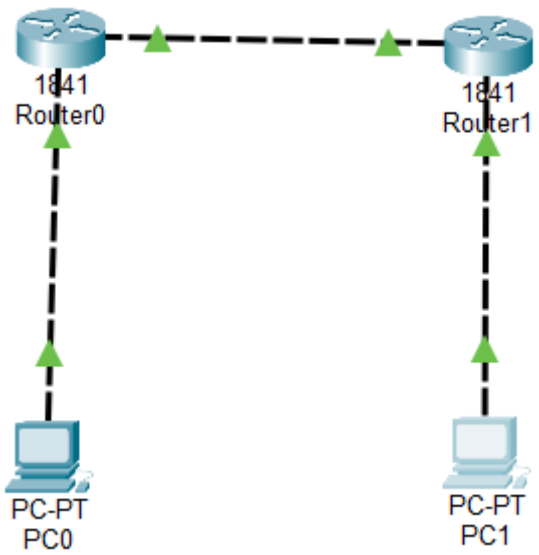
Q. Demonstrate the RIP Routing in Packet Tracer.

Aim: To configure and demonstrate RIP dynamic routing protocol between two networks using two routers in Packet Tracer.

Steps:

1. In packet tracer place 2 PCs and 2 Routers.
2. Connect Router0 with PC0 via FastEthernet0/0 using cross-copper cord and Router1 with PC1 via FastEthernet0/0 using cross-copper cord.
3. Connect both routers via FastEthernet0/1 using cross-copper cord.
4. Set the IP address and default gateway of both the PCs in IP configuration of both PCs respectively.
5. Configure RIP version 2 routing protocol on both routers.

Topology :



Code:

Router0 Configuration:

Router>enable

```
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface FastEthernet0/0
Router0(config-if)#ip address 192.168.10.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#interface FastEthernet0/1
Router0(config-if)#ip address 10.10.10.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#router rip
Router0(config-router)#version 2
Router0(config-router)#network 192.168.10.0
Router0(config-router)#network 10.10.10.0
Router0(config-router)#no auto-summary
Router0(config-router)#exit
Router0(config)#exit
Router0#copy running-config startup-config
```

Router1 configuration:

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface FastEthernet0/1
Router1(config-if)#ip address 10.10.10.2 255.255.255.0
```

Router1(config-if)#no shutdown

Router1(config-if)#exit

Router1(config)#interface FastEthernet0/0

Router1(config-if)#ip address 172.16.20.1 255.255.255.0

Router1(config-if)#no shutdown

Router1(config-if)#exit

Router1(config)#router rip

Router1(config-router)#version 2

Router1(config-router)#network 10.10.10.0

Router1(config-router)#network 172.16.20.0

Router1(config-router)#no auto-summary

Router1(config-router)#exit

Router1(config)#exit

Router1#copy running-config startup-config

PC0 Configuration:

- IP Address: 192.168.10.2
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.10.1

PC1 Configuration:

- IP Address: 172.16.20.2
- Subnet Mask: 255.255.255.0
- Default Gateway: 172.16.20.1

PCO Command Prompt :

C:\>ipconfig

C:\>ping 192.168.10.1

C:\>ping 10.10.10.1

C:\>ping 10.10.10.2

C:\>ping 172.16.20.1

C:\>ping 172.16.20.2

PCO Command Prompt :

C:\>ipconfig

C:\>ping 172.16.20.1

C:\>ping 10.10.10.2

C:\>ping 10.10.10.1

C:\>ping 192.168.10.1

C:\>ping 192.168.10.2

Router Verification Commands:

Router0#show ip route

Router0#show ip protocols

Router0#show ip rip database

Router1#show ip route

Router1#show ip protocols

OUTPUT :

PC0 OUTPUT :

```
C:\>ping 172.16.20.2

Pinging 172.16.20.2 with 32 bytes of data:

Reply from 172.16.20.2: bytes=32 time=1ms TTL=126
Reply from 172.16.20.2: bytes=32 time<1ms TTL=126
Reply from 172.16.20.2: bytes=32 time<1ms TTL=126
Reply from 172.16.20.2: bytes=32 time<1ms TTL=126

Ping statistics for 172.16.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC1 OUTPUT :

```
C:\>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.10.2: bytes=32 time=1ms TTL=126
Reply from 192.168.10.2: bytes=32 time<1ms TTL=126
Reply from 192.168.10.2: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

IP ROUTE OUTPUT :

```
router0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, FastEthernet0/1
     172.16.0.0/24 is subnetted, 1 subnets
R       172.16.20.0 [120/1] via 10.10.10.2, 00:00:13, FastEthernet0/1
C    192.168.10.0/24 is directly connected, FastEthernet0/0
```

RIP PROTOCOL OUTPUT :

```
router0#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 9 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 2, receive 2
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       22
  FastEthernet0/1       22
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
          10.0.0.0
          192.168.10.0
Passive Interface(s):
Routing Information Sources:
          Gateway         Distance      Last Update
          10.10.10.2            120      00:00:02
Distance: (default is 120)
```


CONCLUSION : The above network topology has been executed successfully and RIP version 2 dynamic routing has been demonstrated.

Q. Demonstrate FTP Server in Packet Tracer.

Aim: To configure an FTP server and test file transfer between server and clients in the network.

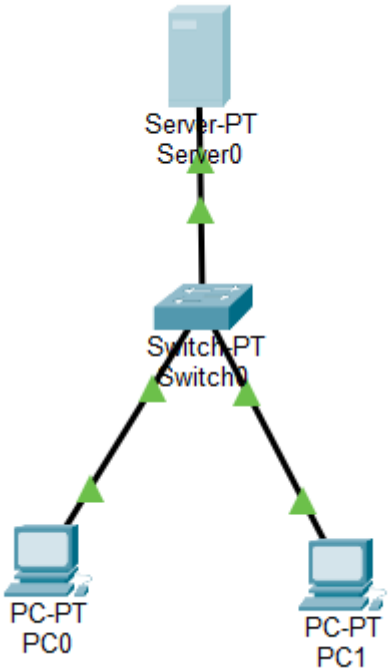
Devices Needed:

- 1 Server (FTP Server)
- 1 Switch
- 2 PCs (FTP Clients)

Connection :

- Connect all devices to the switch using copper straight-through cables

Topology:



Code:

Server Configuration (FTP):

1. Click on Server → Services tab → FTP
2. FTP Server Settings:
    - FTP Service: ON

3. Add User Account:
    - Username: admin
    - Password: cisco
    - Check all permissions: Read/Write/List/Delete/Rename
    - Click Add

Server IP Configuration:

- IP Address: 192.168.1.5
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1

PC1 Configuration:

- IP Address: 192.168.1.10
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1

PC2 Configuration:

- IP Address: 192.168.1.11
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1

Create Test File on PC1

1. Click on PC1 → Desktop tab → Text Editor
2. Type: This is a test file for FTP transfer
3. Save as: upload_test.txt

Verification: In PC Command Prompt

PC1 FTP Commands:

C:\>ftp 192.168.1.5

Username: admin

Password: cisco

ftp> dir

ftp> put upload_test.txt

ftp> dir

ftp> get welcome.txt

ftp> delete notes.txt

ftp> quit

PC2 FTP Commands:

C:\>ftp 192.168.1.5

Username: admin

Password: cisco

ftp> dir

ftp> get upload_test.txt

ftp> rename upload_test.txt downloaded_file.txt

ftp> dir

ftp> quit

Ping Test (Both PCs):

C:\>ping 192.168.1.5

Output:

PC1 FTP Session:

```
C:\>ftp 192.168.1.5
Trying to connect...192.168.1.5
Connected to 192.168.1.5
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:*****
230- Logged in
(passive mode On)
ftp>dir
```

```
33  : notes_text.txt                          24
34  : pt1000-i-mz.122-28.bin                  5571584
35  : pt3000-i6q412-mz.121-22.EA4.bin         3117390
36  : upload_text.txt                         37
37  : welcome_text.txt                        26
```



```
ftp>put upload_text.txt

Writing file upload_text.txt to 192.168.1.5:
File transfer in progress...

[Transfer complete - 36 bytes]

36 bytes copied in 0.021 secs (1714 bytes/sec)
ftp>put welcome_text
```

```
ftp>dir

Listing /ftp directory from 192.168.1.5:
0    : asa842-k8.bin                                    5571584
1    : asa923-k8.bin                                    30468096
2    : c1841-advipservicesk9-mz.124-15.T1.bin           33591768
3    : c1841-ipbase-mz.123-14.T7.bin                    13832032
4    : c1841-ipbasek9-mz.124-12.bin                     16599160
5    : c1900-universalk9-mz.SPA.155-3.M4a.bin           33591768
6    : c2600-advipservicesk9-mz.124-15.T1.bin           33591768
7    : c2600-i-mz.122-28.bin                            5571584
8    : c2600-ipbasek9-mz.124-8.bin                      13169700
9    : c2800nm-advipservicesk9-mz.124-15.T1.bin         50938004
10   : c2800nm-advipservicesk9-mz.151-4.M4.bin          33591768
11   : c2800nm-ipbase-mz.123-14.T7.bin                  5571584
12   : c2800nm-ipbasek9-mz.124-8.bin                    15522644
13   : c2900-universalk9-mz.SPA.155-3.M4a.bin           33591768
14   : c2950-i6q412-mz.121-22.EA4.bin                   3058048
15   : c2950-i6q412-mz.121-22.EA8.bin                   3117390
16   : c2960-lanbase-mz.122-25.FX.bin                   4414921
17   : c2960-lanbase-mz.122-25.SEE1.bin                 4670455
18   : c2960-lanbasek9-mz.150-2.SE4.bin                 4670455
19   : c3560-advipservicesk9-mz.122-37.SE1.bin          8662192
20   : c3560-advipservicesk9-mz.122-46.SE.bin           10713279
21   : c800-universalk9-mz.SPA.152-4.M4.bin             33591768
22   : c800-universalk9-mz.SPA.154-3.M6a.bin            83029236
23   : cat3k_caa-universalk9.16.03.02.SPA.bin           505532849
24   : cgr1000-universalk9-mz.SPA.154-2.CG              159487552
25   : cgr1000-universalk9-mz.SPA.156-3.CG              184530138
26   : ie9k_iosxe.17.09.04.SPA.bin                      596133776
27   : ir800-universalk9-bundle.SPA.156-3.M.bin         160968869
28   : ir800-universalk9-mz.SPA.155-3.M                 61750062
29   : ir800-universalk9-mz.SPA.156-3.M                 63753767
30   : ir800_yocto-1.7.2.tar                            2877440
31   : ir800_yocto-1.7.2_python-2.7.3.tar               6912000
32   : ir8340-mono-universalk9_iot.17.08.01a.SPA.pkg    685645824
33   : notes_text.txt                                   24
34   : pt1000-i-mz.122-28.bin                           5571584
35   : pt3000-i6q412-mz.121-22.EA4.bin                  3117390
36   : upload_text.txt                                  36
37   : welcome.txt                                      25
38   : welcome_text.txt                                 26
```

```
ftp>get welcome.txt

Reading file welcome.txt from 192.168.1.5:
File transfer in progress...

[Transfer complete - 25 bytes]

25 bytes copied in 0 secs
```

```
ftp>delete notes.txt

Deleting file notes.txt from 192.168.1.5: ftp>
[Deleted file notes.txt successfully ]
```

```
ftp>quit

221- Service closing control connection.
```

Ping Test Results:

```
C:\>ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data:

Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time=1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC2 FTP Session:

```
C:\>ftp 192.168.1.5
Trying to connect...192.168.1.5
Connected to 192.168.1.5
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:*****
230- Logged in
(passive mode On)
ftp>dir

Listing /ftp directory from 192.168.1.5:
```

```
31   : ir800_yocto-1.7.2_python-2.7.3.tar                  6912000
32   : ir8340-mono-universalk9_iot.17.08.01a.SPA.pkg       685645824
33   : notes_text.txt                                      24
34   : pt1000-i-mz.122-28.bin                              5571584
35   : pt3000-i6q412-mz.121-22.EA4.bin                     3117390
36   : upload_text.txt                                     36
37   : welcome.txt                                         25
38   : welcome_text.txt                                    26
```

```
ftp>get upload_text.txt

Reading file upload_text.txt from 192.168.1.5:
File transfer in progress...

[Transfer complete - 37 bytes]

37 bytes copied in 0 secs
```

```
ftp>rename upload_text.txt downloaded_text.txt

Renaming upload_text.txt
ftp>
[OK Renamed file successfully from upload_text.txt to downloaded_text.txt]
```

```
ftp>dir

Listing /ftp directory from 192.168.1.5:
0    : asa842-k8.bin                                    5571584
1    : asa923-k8.bin                                    30468096
2    : c1841-advipservicesk9-mz.124-15.T1.bin           33591768
3    : c1841-ipbase-mz.123-14.T7.bin                    13832032
```

```
25   : cgr1000-universalk9-mz.SPA.156-3.CG              184530138
26   : downloaded_text.txt                              37
27   : ie9k_iosxe.17.09.04.SPA.bin                      596133776
28   : ir800-universalk9-bundle.SPA.156-3.M.bin         160968869
29   : ir800-universalk9-mz.SPA.155-3.M                 61750062
30   : ir800-universalk9-mz.SPA.156-3.M                 63753767
31   : ir800_yocto-1.7.2.tar                            2877440
32   : ir800_yocto-1.7.2_python-2.7.3.tar               6912000
33   : ir8340-mono-universalk9_iot.17.08.01a.SPA.pkg    685645824
34   : notes_text.txt                                   24
35   : pt1000-i-mz.122-28.bin                           5571584
36   : pt3000-i6q4l2-mz.121-22.EA4.bin                  3117390
37   : welcome.txt                                      25
38   : welcome_text.txt                                 26
```

```
ftp>quit

221- Service closing control connection.
```

Ping Test Results:

```
C:\>ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data:

Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Conclusion: The FTP server has been configured successfully.

PARTICAL NO. 4

STUDY OF CRYPTOGRAPHY -SUBSTITUTION TECHNIQUES

Aim: To study and implement Substitution Techniques in Cryptography using Python to perform encryption and decryption of a message

Program:

```
def caesar_cipher(text, shift):
    result = ""
    for char in text:
        if char.isalpha():
            base = 'A' if char.isupper() else 'a'
            result += chr((ord(char) - ord(base) + shift) % 26 + ord(base))
        else:
            result += char
    return result

plaintext = input("Enter plaintext: ")
shift = int(input("Enter shift value: "))

encrypted_text = caesar_cipher(plaintext, shift)
print("\nEncrypted Text:", encrypted_text)

decrypted_text = caesar_cipher(encrypted_text, -shift)
print("Decrypted Text:", decrypted_text)
```

OUTPUT:

```
================== RESTART: C:/Users/VAISHNAVI/CASER_CIPHER.PY ==================
Enter plaintext: VAISHNAVI
Enter shift value: 4

Encrypted Text: ZEMWLREZM
Decrypted Text: VAISHNAVI
```

CONCLUSION: The above code has been executed successfully.

PARTICAL NO. 5

Study of Symmetric Key Cryptography (DES Algorithm)

AIM: To study and implement Symmetric Key in Cryptography (DSE Algorithm) using Python to perform encryption and decryption of a message

PROGRAM:

```
from Crypto.Cipher import DES

from Crypto.Util.Padding import pad, unpad


key = b'8bytekey'

cipher = DES.new(key, DES.MODE_ECB)


plaintext = input("Enter message to encrypt: ").encode()


padded_text = pad(plaintext, DES.block_size)

encrypted_text = cipher.encrypt(padded_text)

print("\nEncrypted (ciphertext):", encrypted_text)


decrypted_text = unpad(cipher.decrypt(encrypted_text), DES.block_size)

print("Decrypted (plaintext):", decrypted_text.decode())
```

OUTPUT:

```
Enter message to encrypt: VAISHNAVI

Encrypted (ciphertext): b'\xb0\xc8\x8e\x9d\x9e\x93\xfak\xf9\xd2x\xd0\xba\xb4\x87t'
Decrypted (plaintext): VAISHNAVI
```

CONCLUSION: The above code has been executed successfully.

Study of Symmetric Key Cryptography (AES Algorithm)

AIM: To study and implement Symmetric Key in Cryptography (ASE Algorithm) using Python to perform encryption and decryption of a message

PROGRAM:

```
from Crypto.Cipher import AES

from Crypto.Random import get_random_bytes

from Crypto.Util.Padding import pad, unpad

key = get_random_bytes(16)

print("Secret Key:", key)


cipher = AES.new(key, AES.MODE_ECB)

plaintext = input("\nEnter message to encrypt: ").encode()


padded_text = pad(plaintext, AES.block_size)

encrypted_text = cipher.encrypt(padded_text)

print("\nEncrypted (ciphertext):", encrypted_text)


decrypted_text = unpad(cipher.decrypt(encrypted_text), AES.block_size)

print("Decrypted (plaintext):", decrypted_text.decode())
```

OUTPUT:

```
Secret Key: b"q'\xa1o\x05;J\xed)\xa8\xb7\x9e\xc3\xda\x1c~"

Enter message to encrypt: HELLO WORLD

Encrypted (ciphertext): b'\xcb\x9a\xa3\x86\x8a\xe8c\x8a/M\xa8%\xd5\xd7_\xf9'
Decrypted (plaintext): HELLO WORLD
```

CONCLUSION: The above code has been executed successfully.

PRACTICAL NO. 6

Study of Assymetric key (DH)

AIM: To study and implement asymmetric key cryptography using Diffie–Hellman (DH) algorithms in Python.

PROGRAM:

```
P = 23  # Prime number

G = 9   # Primitive root


a = 4  # Alice's private key

b = 3  # Bob's private key


A = (G ** a) % P

B = (G ** b) % P


key1 = (B ** a) % P  # Alice's secret key

key2 = (A ** b) % P  # Bob's secret key


print("Publicly Shared Values:")

print("P:", P, "G:", G)


print("Public Key of Alice (A):", A)

print("Public Key of Bob (B):", B)


print("\nSecret Keys:")
```

```
print("Alice's Secret Key:", key1)

print("Bob's Secret Key:", key2)
```

OUTPUT:

```
Publicly Shared Values:
P: 23 G: 9
Public Key of Alice (A): 6
Public Key of Bob (B): 16

Secret Keys:
Alice's Secret Key: 9
Bob's Secret Key: 9
```


CONCLUSION: The above code has been executed successfully.

Study of Assymetric key (RSA)

AIM: To study and implement asymmetric key cryptography using Diffie–Hellman RSA algorithms in Python

PROGRAM:

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

p = 7
q = 17
n = p * q
phi = (p - 1) * (q - 1)
e = 5
while gcd(e, phi) != 1:
    e += 1
d = pow(e, -1, phi)

print("Public Key (e, n):", (e, n))
print("Private Key (d, n):", (d, n))

msg = int(input("Enter message (as number): "))
cipher = pow(msg, e, n)
print("Encrypted Message:", cipher)
```

```
decrypted = pow(cipher, d, n)

print("Decrypted Message:", decrypted)
```

OUTPUT:

```
Public Key (e, n): (5, 119)
Private Key (d, n): (77, 119)
Enter message (as number): 25
Encrypted Message: 9
Decrypted Message: 25
```

CONCLUSION: The above code has been executed successfully.

PRACTICAL NO. 7

Study of MD5 Algorithm

Aim: To study and implement the MD5 (Message Digest 5) hashing algorithm.

PROGRAM:

```
import hashlib

message = input("Enter a message to hash using MD5: ")

encoded_message = message.encode()

md5_hash = hashlib.md5(encoded_message)

digest = md5_hash.hexdigest()

print("\nOriginal Message:", message)

print("MD5 Hash Value:", digest)
```

OUTPUT:

```
Enter a message to hash using MD5: HELLO WORLD

Original Message: HELLO WORLD
MD5 Hash Value: 035906f405c32b287fa1ddd767f1edda
```

CONCLUSION: The above code has been executed successfully.

PRACTICAL NO. 8

Study of Hash Function – RSHash

AIM: To study and implement the RSHash algorithm in python.

PROGRAM:

```
def RSHash(string):
    a = 63689
    b = 378551
    hash_value = 0

    for ch in string:
        hash_value = hash_value * a + ord(ch)
        a = a * b

    return hash_value & 0x7FFFFFFF
message = input("Enter a message to hash using RSHash: ")

hash_result = RSHash(message)

print("\nOriginal Message:", message)
print("RSHash Value:", hash_result)
```

OUTPUT:

```
Enter a message to hash using RSHash: my name is vaishnavi

Original Message: my name is vaishnavi
RSHash Value: 864809090
```

CONCLUSION: The above code has been executed successfully.

PRACTICAL NO. 9

To create , export and validate a digital certificate

AIM: To write a code to Create, Export and Validate a digital Certificate.

STEPS:

1. Download and Install opensol
2. Add its bin path in environmental variable
3. Open command prompt

PROGRAM:

C:\User\Vaishnavi\Desktop>openssl req -x509 -days 365 -newkey rsa:2048 -keyout private-key.pem -out certificate.pem

Enter PEM Pass: 123456

Country Name (2 letter code) []: IN

State or Province Name (full name) []: Maharashtra

Locality Name []: Mumbai

Organization Name []: SIWS

Organizational Unit Name []: IT

Common Name []: TMD

Email Address []: vaishnavithevar@gmail.com

C:\User\Vaishnavi\Desktop> openssl pkcs12 -export -in certificate.pem -inkey private-key.pem -out TMD.pfx

Enter Export Password: 123456

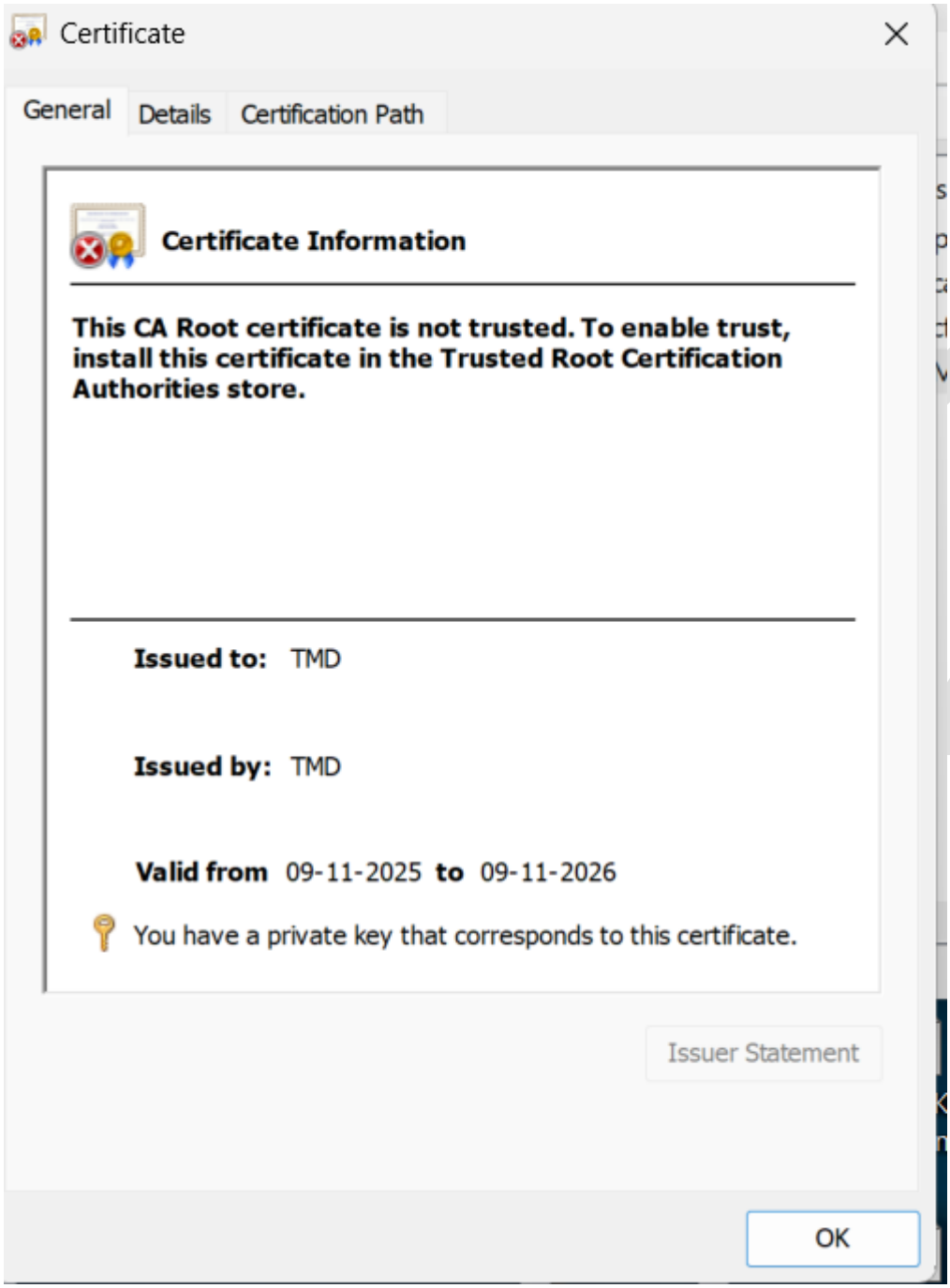
Verifying - Enter Export Password: 123456

C:\User\Vaishnavi\Desktop> openssl pkcs12 -in TMD.pfx -clcerts -nokeys -out public-key.pem

Enter Import Password: 123456

Import the certificate

OUTPUT:



CONCLUSION: The above creation is done successfully.