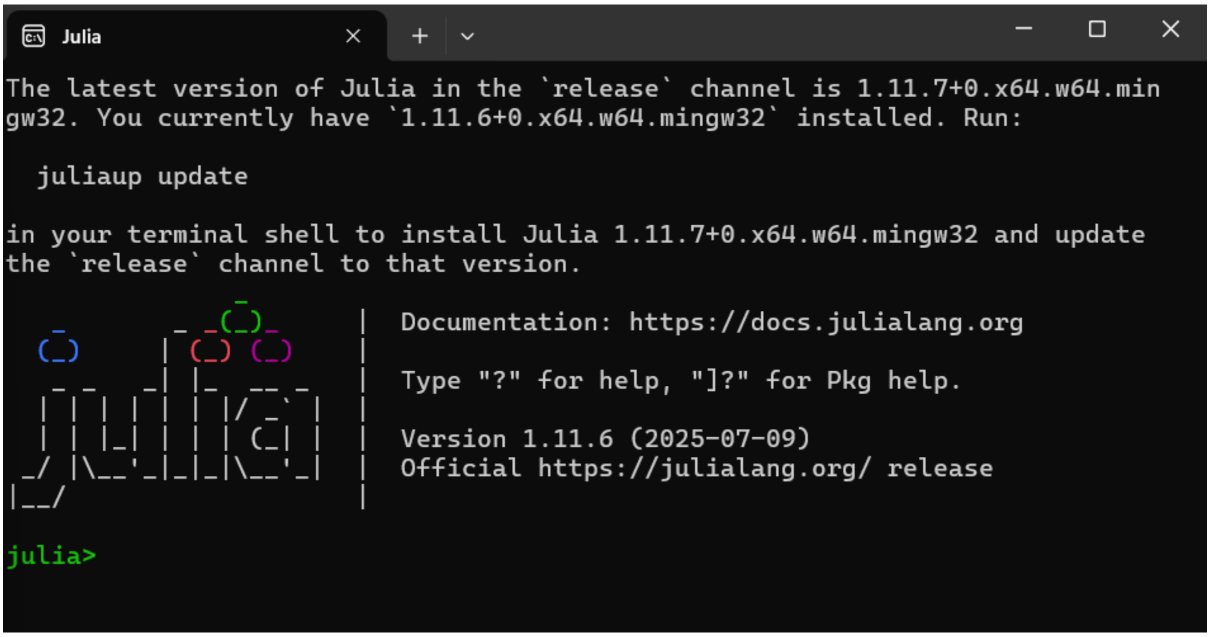
ALGORITHM FOR OPTIMIZATION

INDEX

PRACTICAL NO. 1

Aim : Implement Contour Plots.

Step 1 : Open Julia



```
The latest version of Julia in the `release` channel is 1.11.7+0.x64.w64.min
gw32. You currently have `1.11.6+0.x64.w64.mingw32` installed. Run:

  juliaup update

in your terminal shell to install Julia 1.11.7+0.x64.w64.mingw32 and update
the `release` channel to that version.
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.11.6 (2025-07-09)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia>
```

Step 2 : Install Package PlotyJS

Command :

```
import Pkg; Pkg.add("PlotlyJS")
```

Step 3 : To implementing counter plot use PlotlyJS

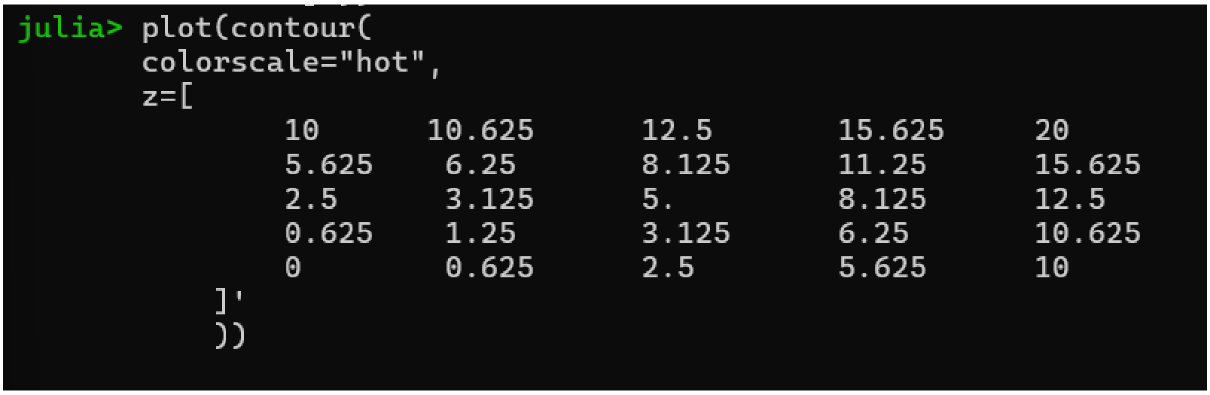
Command :

```
using PlotlyJS
```

Step 4 : Command to implement the Plot

Command :

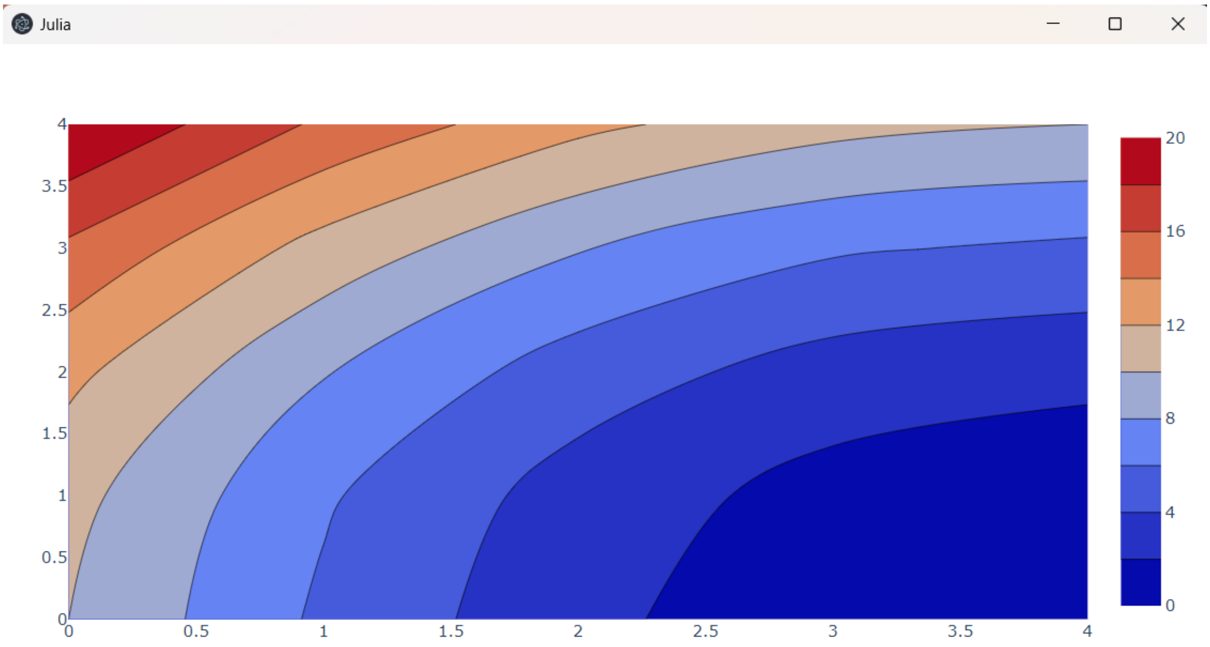
```
plot(countour(
colorscale=”hot”,
z=[
        10          10.625      12.5        15.625      20
        5.625       6.25        8.125       11.25       15.625
        2.5         3.125       5.0         8.125       12.5
        0.625       1.25        3.125       6.25        10.625
        0           0.625       2.5         5.625       10
        ]’
))
```



```
julia> plot(contour(
       colorscale="hot",
       z=[
               10       10.625      12.5       15.625      20
               5.625     6.25       8.125      11.25       15.625
               2.5       3.125      5.         8.125       12.5
               0.625     1.25       3.125      6.25        10.625
               0         0.625      2.5        5.625       10
           ]'
           ))
```

- ➢ After pess the enter ,Plot is Drow in new Julia page

Output :



PRACTICAL NO . 2

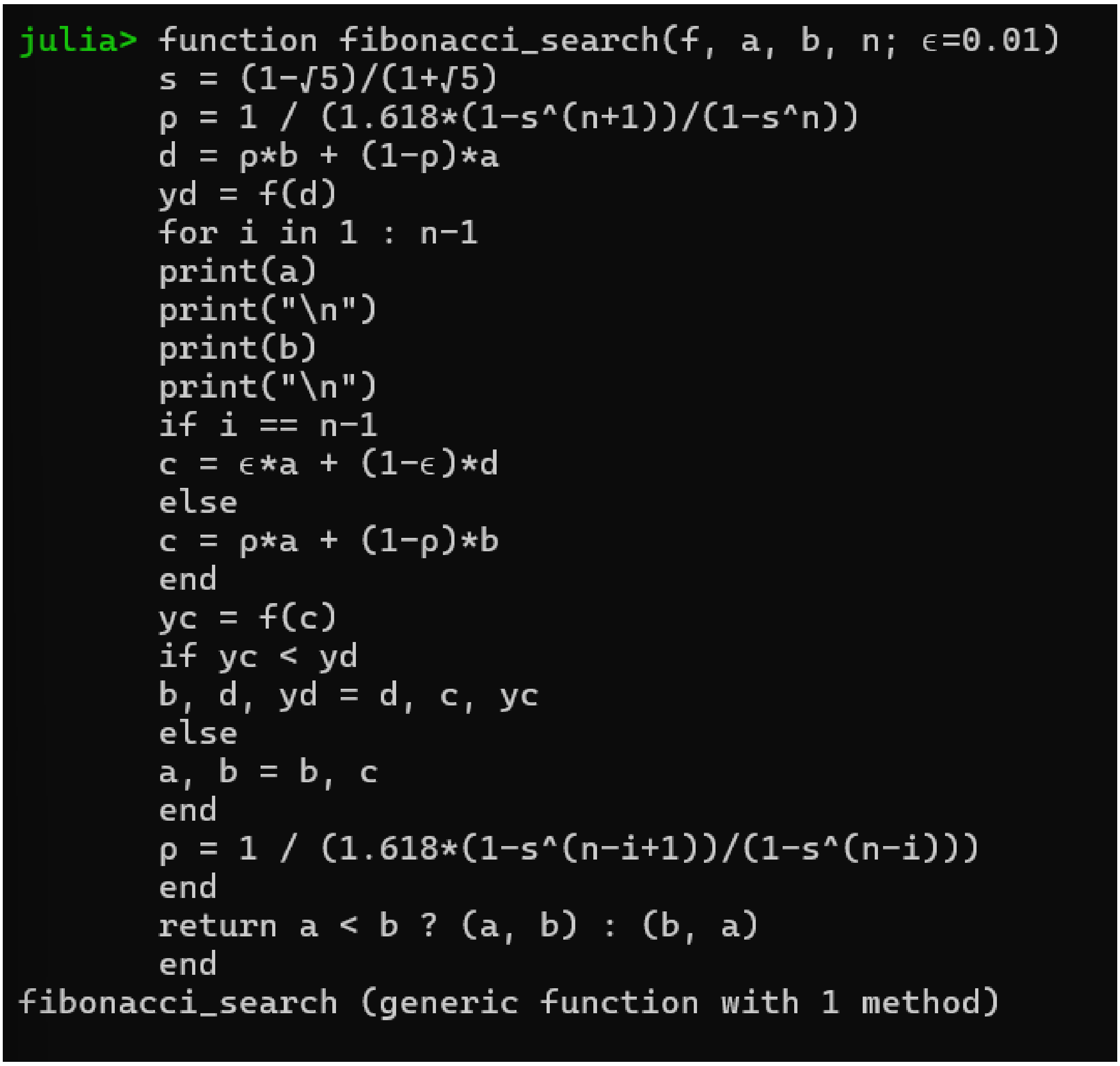
Aim : Implement Fibonacci Search.

Step 1 : Open Julia

Step 2 : Write Fibonacci Search Code

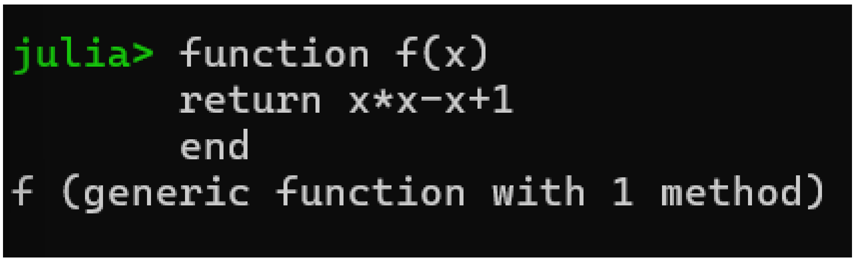
Code :

```
function fibonacci_search(f, a, b, n; ϵ=0.01)
s = (1-√5)/(1+√5)
ρ = 1 / (1.618*(1-s^(n+1))/(1-s^n))
d = ρ*b + (1-ρ)*a
yd = f(d)
for i in 1 : n-1
print(a)
print("\n")
print(b)
print("\n")
if i == n-1
c = ϵ*a + (1-ϵ)*d
else
c = ρ*a + (1-ρ)*b
end
yc = f(c)
if yc < yd
b, d, yd = d, c, yc
else
a, b = b, c
end
ρ = 1 / (1.618*(1-s^(n-i+1))/(1-s^(n-i)))
end
return a < b ? (a, b) : (b, a)
end
```



```
julia> function fibonacci_search(f, a, b, n; ϵ=0.01)
       s = (1-√5)/(1+√5)
       ρ = 1 / (1.618*(1-s^(n+1))/(1-s^n))
       d = ρ*b + (1-ρ)*a
       yd = f(d)
       for i in 1 : n-1
       print(a)
       print("\n")
       print(b)
       print("\n")
       if i == n-1
       c = ϵ*a + (1-ϵ)*d
       else
       c = ρ*a + (1-ρ)*b
       end
       yc = f(c)
       if yc < yd
       b, d, yd = d, c, yc
       else
       a, b = b, c
       end
       ρ = 1 / (1.618*(1-s^(n-i+1))/(1-s^(n-i)))
       end
       return a < b ? (a, b) : (b, a)
       end
fibonacci_search (generic function with 1 method)
```

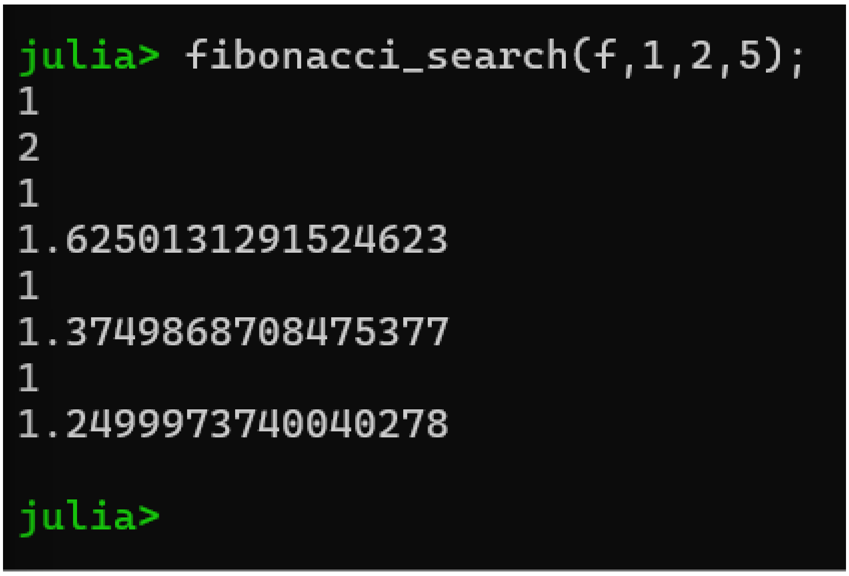
```
function f(x)

return x*x-x+1

end
```



```
julia> function f(x)
       return x*x-x+1
       end
f (generic function with 1 method)
```

Output :

fibonacci_search(f,1,2,5);



```
julia> fibonacci_search(f,1,2,5);
1
2
1
1.6250131291524623
1
1.3749868708475377
1
1.2499973740040278

julia>
```

PRACTICAL NO. 3

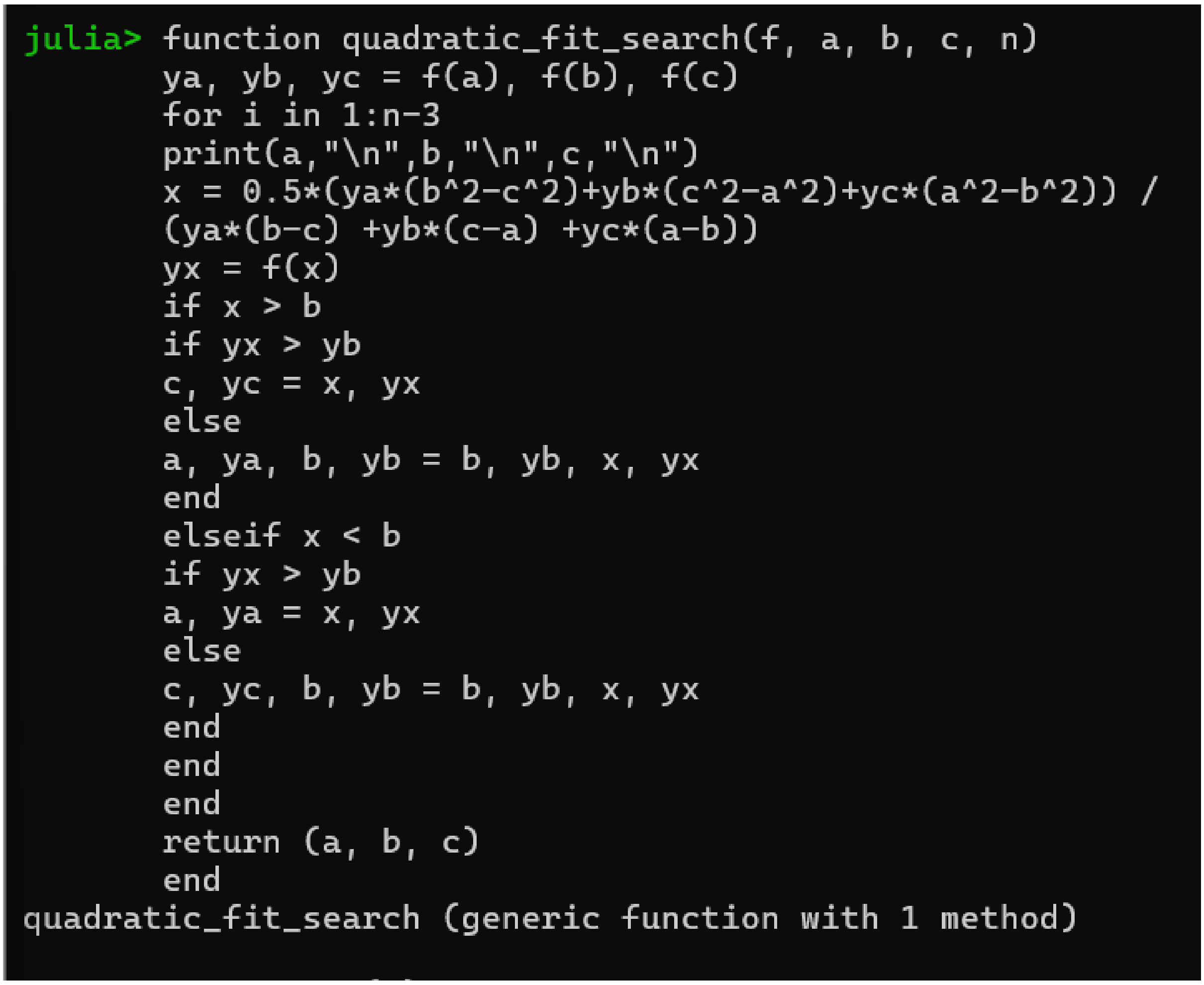
Aim : Implement Quadratic Fit Search.

Step 1 : Open Julia

Step 2 : Write a code on Julia for Quadratic Fit Search

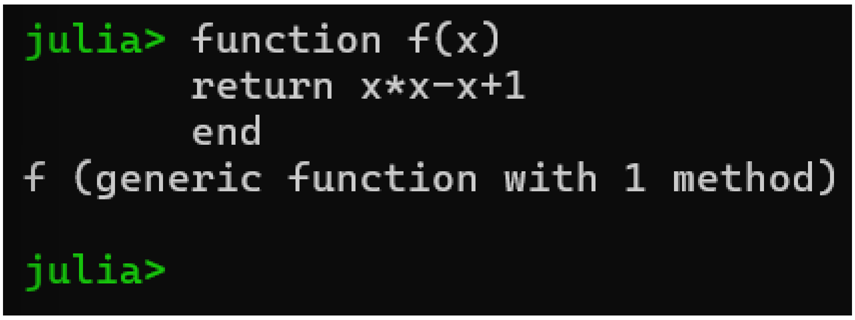
Code :

```
function quadratic_fit_search(f, a, b, c, n)
ya, yb, yc = f(a), f(b), f(c)
for i in 1:n-3
print(a,"\n",b,"\n",c,"\n")
x = 0.5*(ya*(b^2-c^2)+yb*(c^2-a^2)+yc*(a^2-b^2)) /
(ya*(b-c) +yb*(c-a) +yc*(a-b))
yx = f(x)
if x > b
if yx > yb
c, yc = x, yx
else
a, ya, b, yb = b, yb, x, yx
end
elseif x < b
if yx > yb
a, ya = x, yx
else
c, yc, b, yb = b, yb, x, yx
end
end
end
return (a, b, c)
end
```



```
julia> function quadratic_fit_search(f, a, b, c, n)
       ya, yb, yc = f(a), f(b), f(c)
       for i in 1:n-3
       print(a,"\n",b,"\n",c,"\n")
       x = 0.5*(ya*(b^2-c^2)+yb*(c^2-a^2)+yc*(a^2-b^2)) /
       (ya*(b-c) +yb*(c-a) +yc*(a-b))
       yx = f(x)
       if x > b
       if yx > yb
       c, yc = x, yx
       else
       a, ya, b, yb = b, yb, x, yx
       end
       elseif x < b
       if yx > yb
       a, ya = x, yx
       else
       c, yc, b, yb = b, yb, x, yx
       end
       end
       end
       return (a, b, c)
       end
quadratic_fit_search (generic function with 1 method)
```

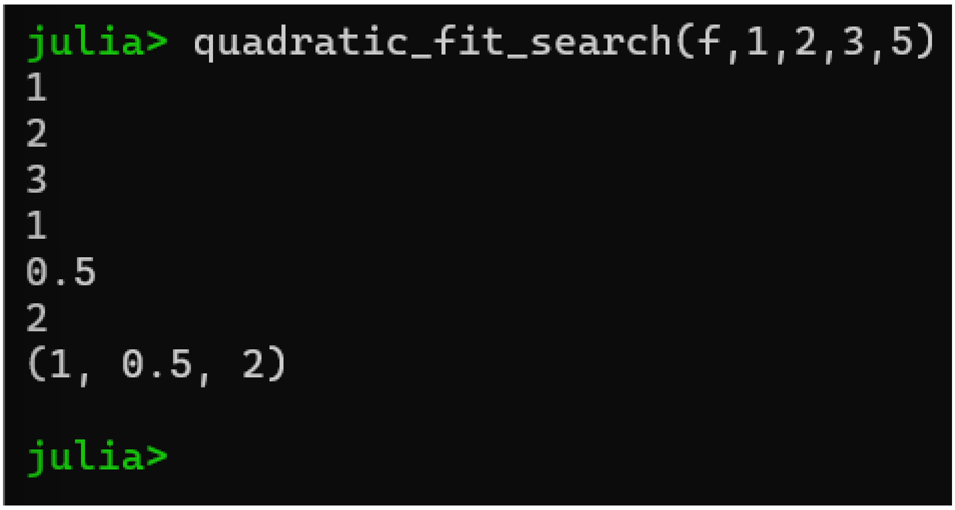
```
function f(x)

return x*x-x+1

end
```



```
julia> function f(x)
       return x*x-x+1
       end
f (generic function with 1 method)

julia>
```

Output :

quadratic_fit_search(f,1,2,3,5)



```
julia> quadratic_fit_search(f,1,2,3,5)
1
2
3
1
0.5
2
(1, 0.5, 2)

julia>
```

PRACTICAL NO. 4

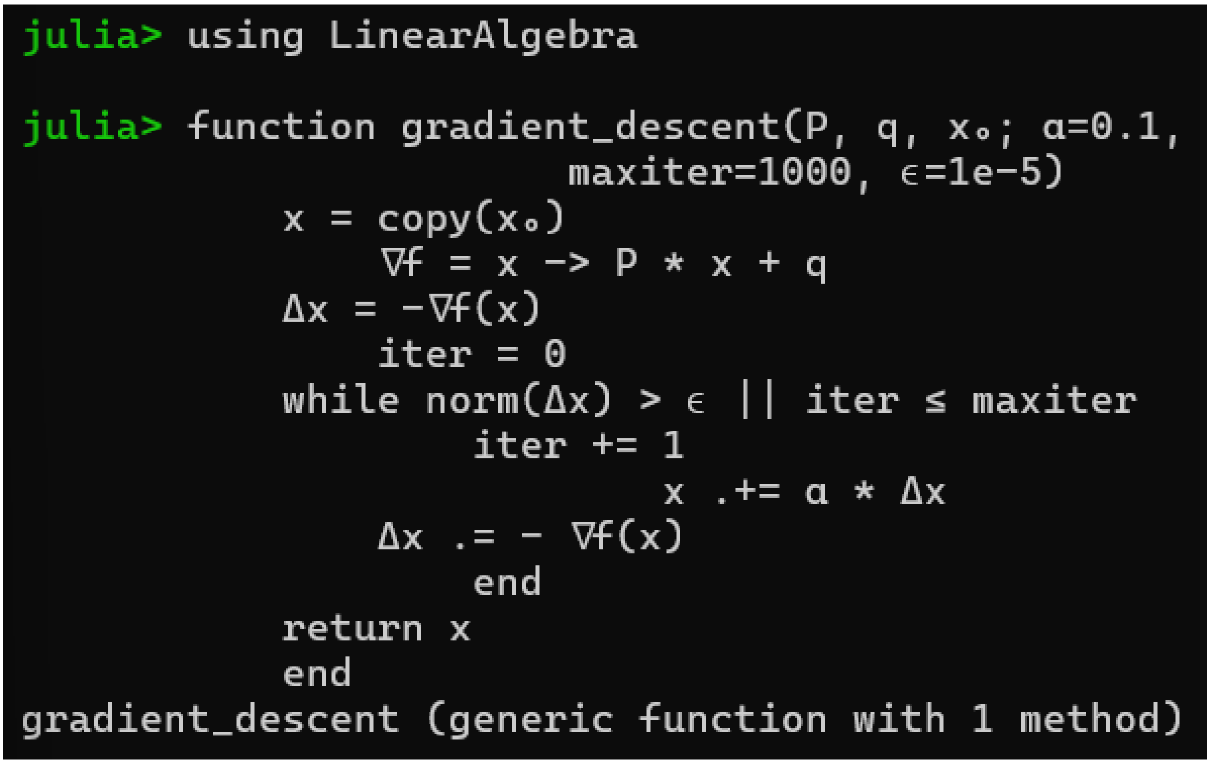
Aim : Implement Gradient Descent.

Step 1 : Open Julia

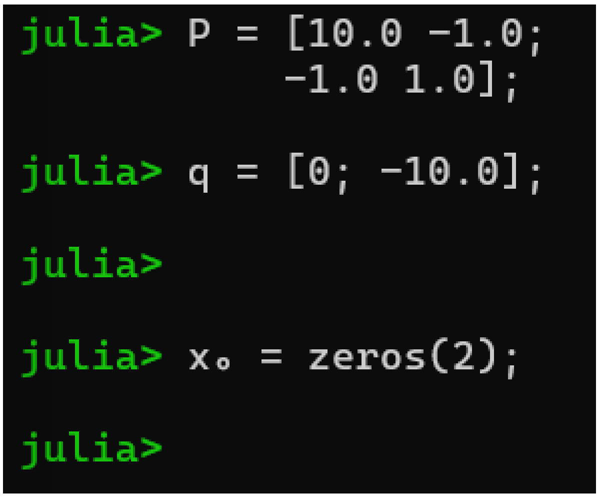
Step 2 : Write a code for Implement Gradient Descent.

Code :

```
using LinearAlgebra


function gradient_descent(P, q, x₀; α=0.1,
maxiter=1000, ϵ=1e-5)
x = copy(x₀)
∇f = x -> P * x + q
Δx = -∇f(x)
iter = 0
while norm(Δx) > ϵ || iter ≤ maxiter
iter += 1
x .+= α * Δx
Δx .= - ∇f(x)
end
return x
end


P = [10.0 -1.0;
    -1.0 1.0];


q = [0; -10.0];


x₀ = zeros(2);
```



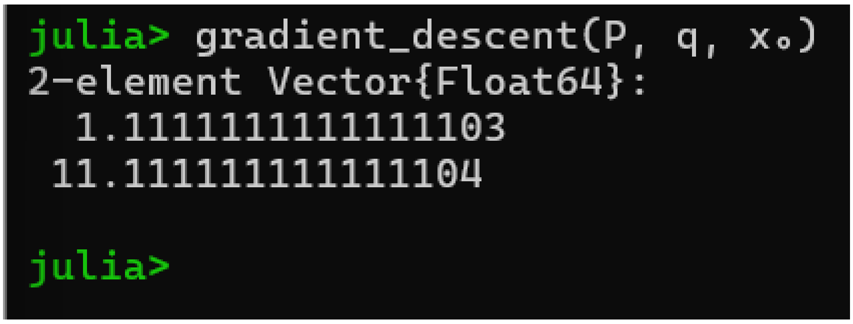
```
julia> using LinearAlgebra

julia> function gradient_descent(P, q, x₀; α=0.1,
                       maxiter=1000, ϵ=1e-5)
           x = copy(x₀)
               ∇f = x -> P * x + q
           Δx = -∇f(x)
               iter = 0
           while norm(Δx) > ϵ || iter ≤ maxiter
                   iter += 1
                           x .+= α * Δx
               Δx .= - ∇f(x)
                   end
           return x
           end
gradient_descent (generic function with 1 method)
```



```
julia> P = [10.0 -1.0;
           -1.0 1.0];

julia> q = [0; -10.0];

julia>

julia> x₀ = zeros(2);

julia>
```

Output :

gradient_descent(P, q, x₀)



```
julia> gradient_descent(P, q, x₀)
2-element Vector{Float64}:
  1.1111111111111103
 11.111111111111104

julia>
```

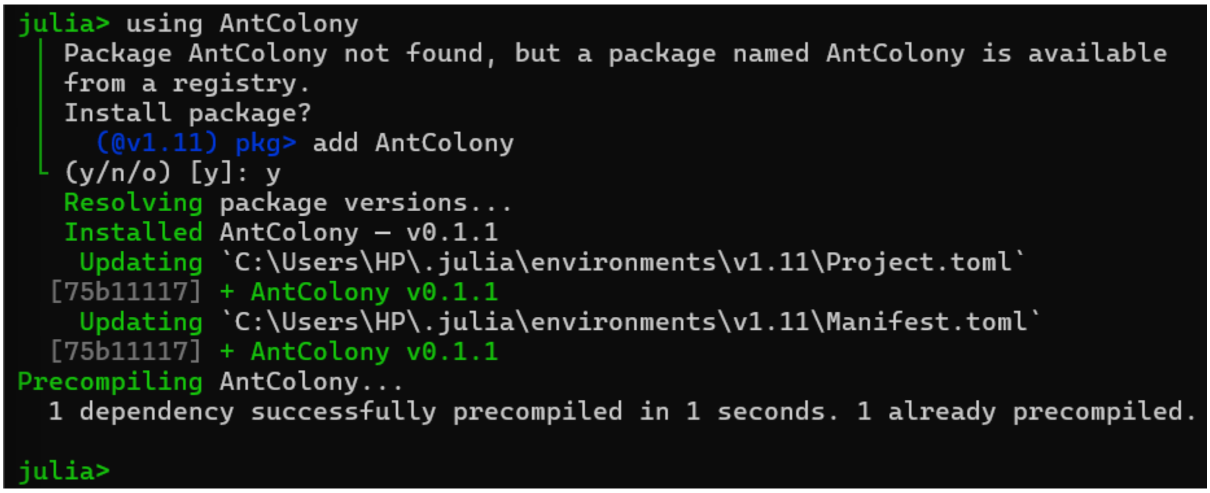
PRACTICAL NO. 5

Aim : Path Finding Using Ant Colony Optimization.

Step 1 : Open Julia

Step 2 : To find ant colony optimization we need to use AntColony Package.

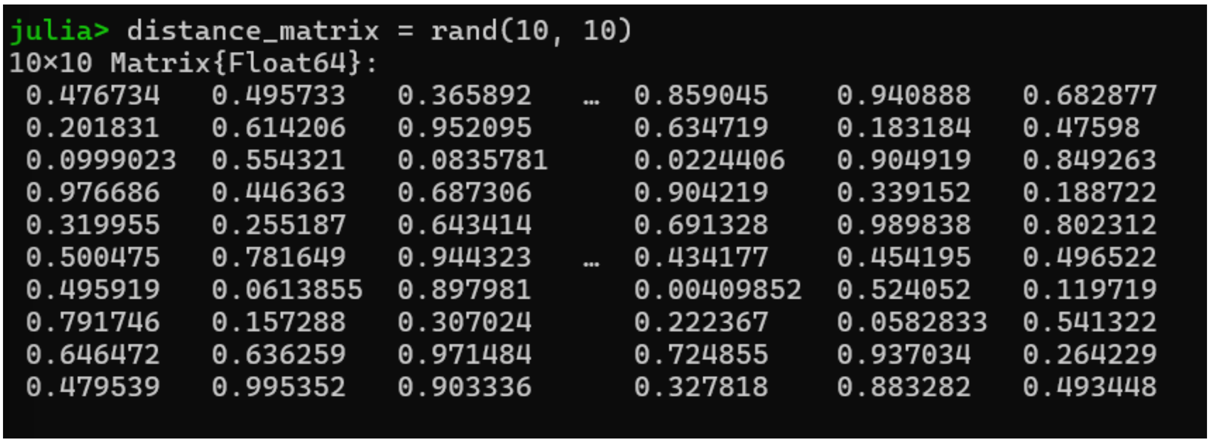
using AntColony



```
julia> using AntColony
 │ Package AntColony not found, but a package named AntColony is available
 │ from a registry.
 │ Install package?
 │   (@v1.11) pkg> add AntColony
 └ (y/n/o) [y]: y
   Resolving package versions...
   Installed AntColony ─ v0.1.1
    Updating `C:\Users\HP\.julia\environments\v1.11\Project.toml`
  [75b11117] + AntColony v0.1.1
    Updating `C:\Users\HP\.julia\environments\v1.11\Manifest.toml`
  [75b11117] + AntColony v0.1.1
Precompiling AntColony...
  1 dependency successfully precompiled in 1 seconds. 1 already precompiled.

julia>
```

Step 3 : Write Commands

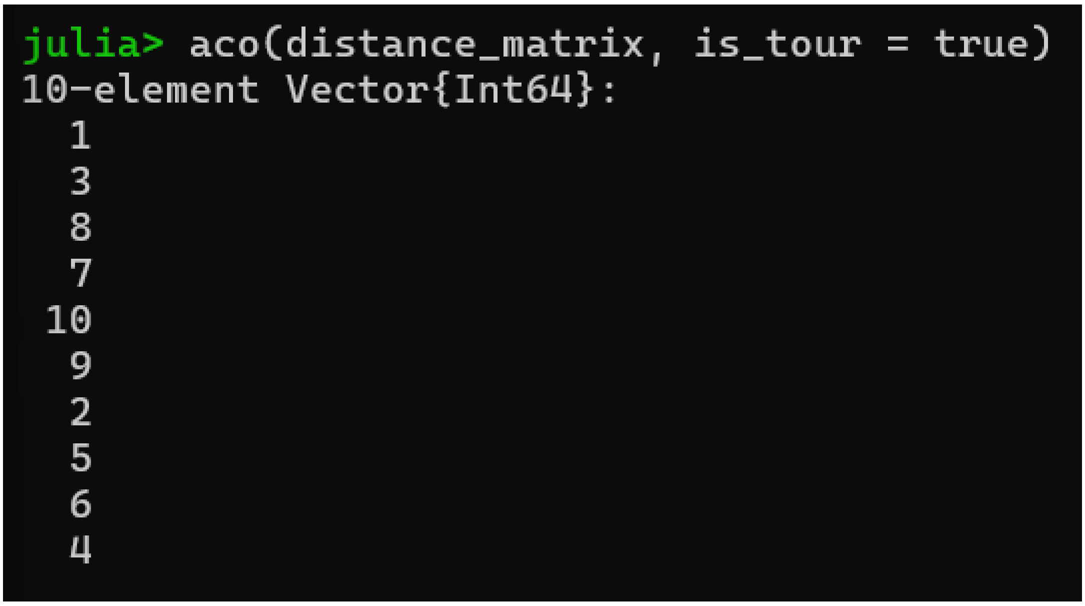
- ➢ distance_matrix = rand(10, 10)



```
julia> distance_matrix = rand(10, 10)
10×10 Matrix{Float64}:
 0.476734   0.495733   0.365892   …  0.859045    0.940888   0.682877
 0.201831   0.614206   0.952095      0.634719    0.183184   0.47598
 0.0999023  0.554321   0.0835781     0.0224406   0.904919   0.849263
 0.976686   0.446363   0.687306      0.904219    0.339152   0.188722
 0.319955   0.255187   0.643414      0.691328    0.989838   0.802312
 0.500475   0.781649   0.944323   …  0.434177    0.454195   0.496522
 0.495919   0.0613855  0.897981      0.00409852  0.524052   0.119719
 0.791746   0.157288   0.307024      0.222367    0.0582833  0.541322
 0.646472   0.636259   0.971484      0.724855    0.937034   0.264229
 0.479539   0.995352   0.903336      0.327818    0.883282   0.493448
```

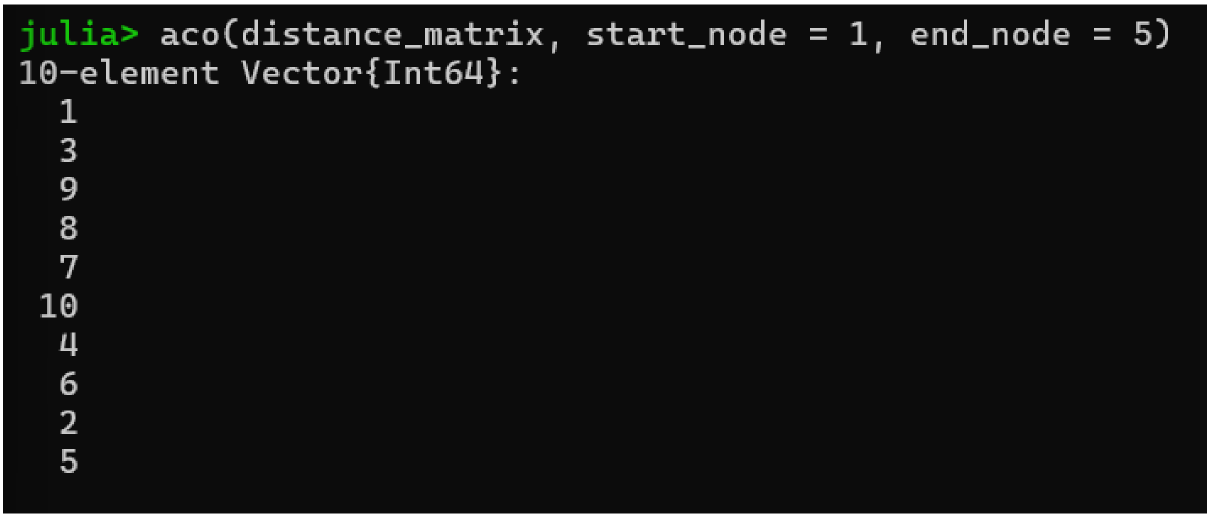
- ➢ Note that distance_matrix[3, 5] = travel distance from node 5 to node 3

- aco(distance_matrix, is_tour = true)



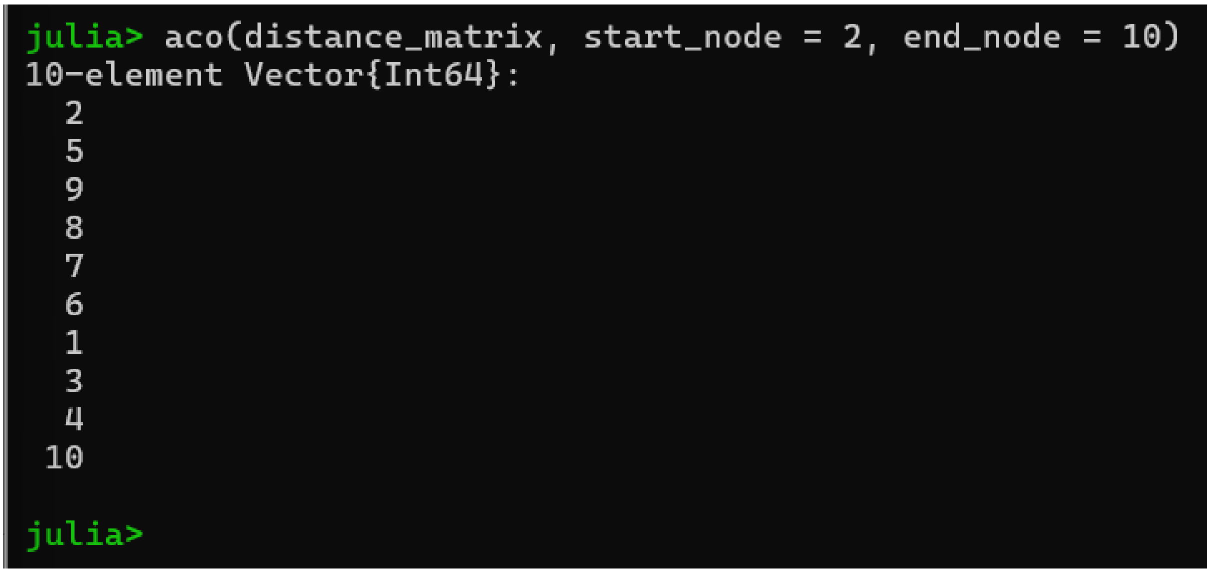
```
julia> aco(distance_matrix, is_tour = true)
10-element Vector{Int64}:
  1
  3
  8
  7
 10
  9
  2
  5
  6
  4
```

- aco(distance_matrix, start_node = 1, end_node = 5)



```
julia> aco(distance_matrix, start_node = 1, end_node = 5)
10-element Vector{Int64}:
  1
  3
  9
  8
  7
 10
  4
  6
  2
  5
```

- aco(distance_matrix, start_node = 2, end_node = 10)



```
julia> aco(distance_matrix, start_node = 2, end_node = 10)
10-element Vector{Int64}:
  2
  5
  9
  8
  7
  6
  1
  3
  4
 10

julia>
```

SOFTWARE DEFINE NETWORKING

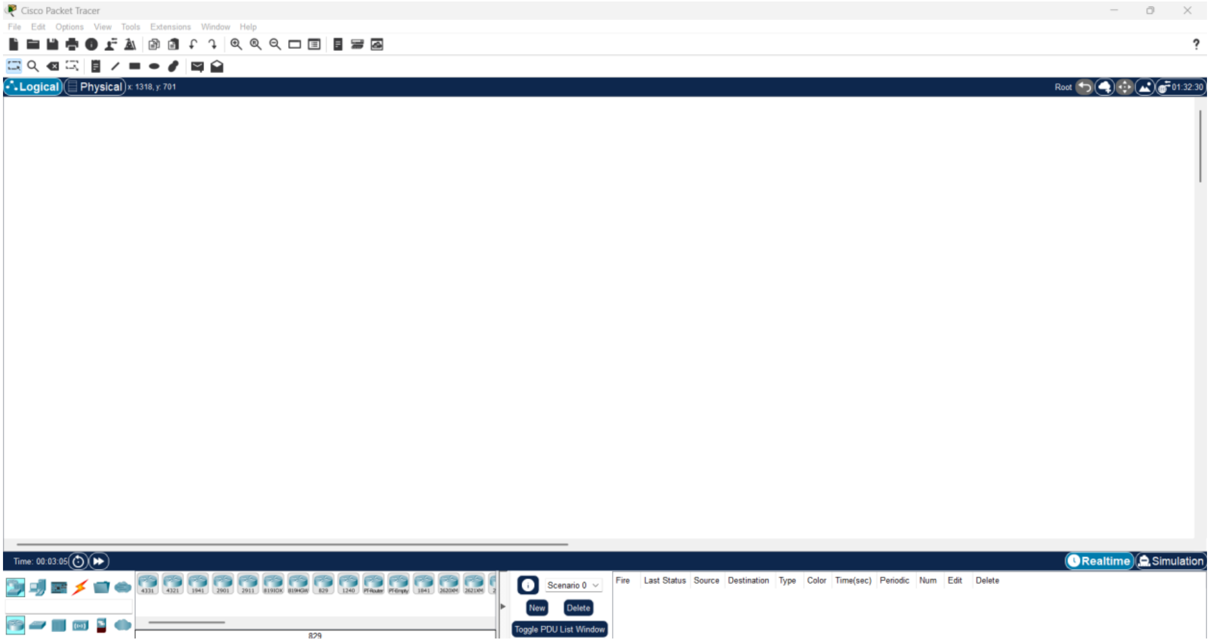
INDEX

PRACTICAL NO. 1

Aim : Perform Basic Commands in Cisco Packet Tracer

Step 1 :

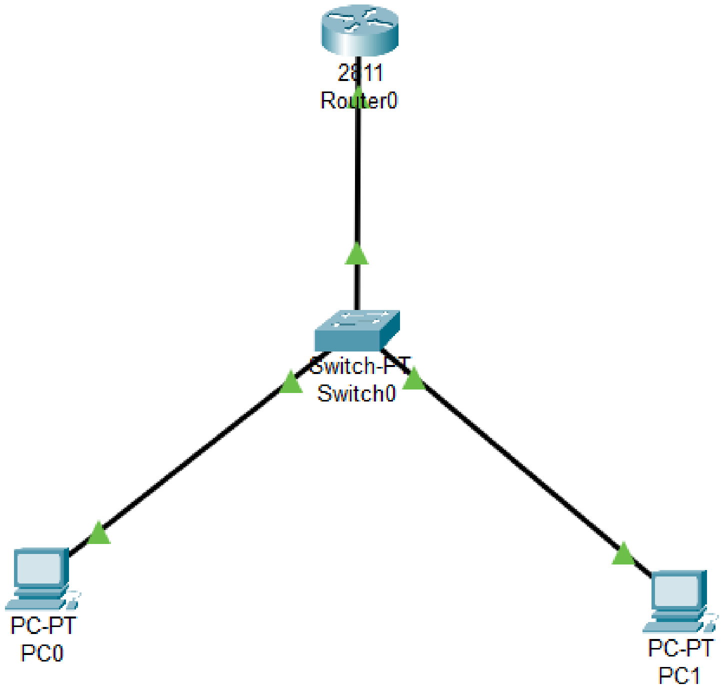
- Open Cisco Packet Tracer



Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical
Physical
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete

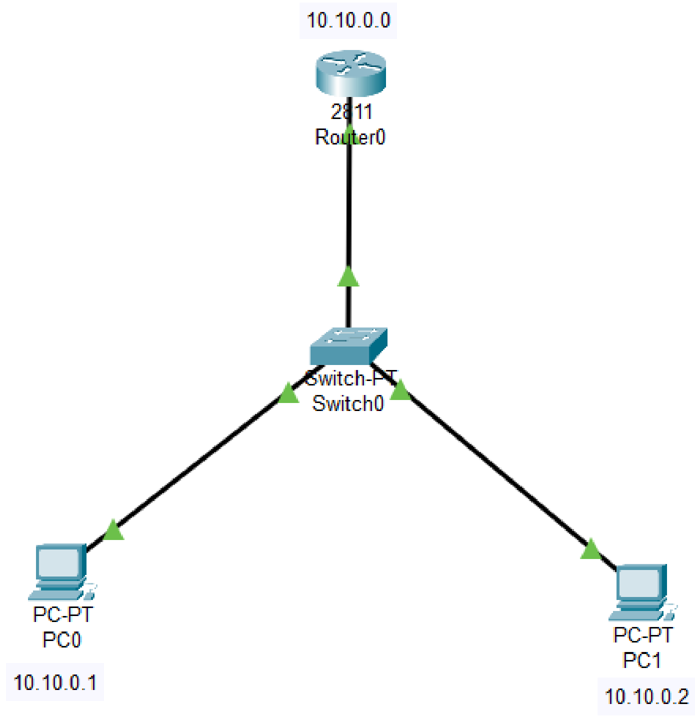
Step 2 :

- Take 2 PC, Switch and Router.
- Make appropriate connection by using “Copper Straight-Through”.



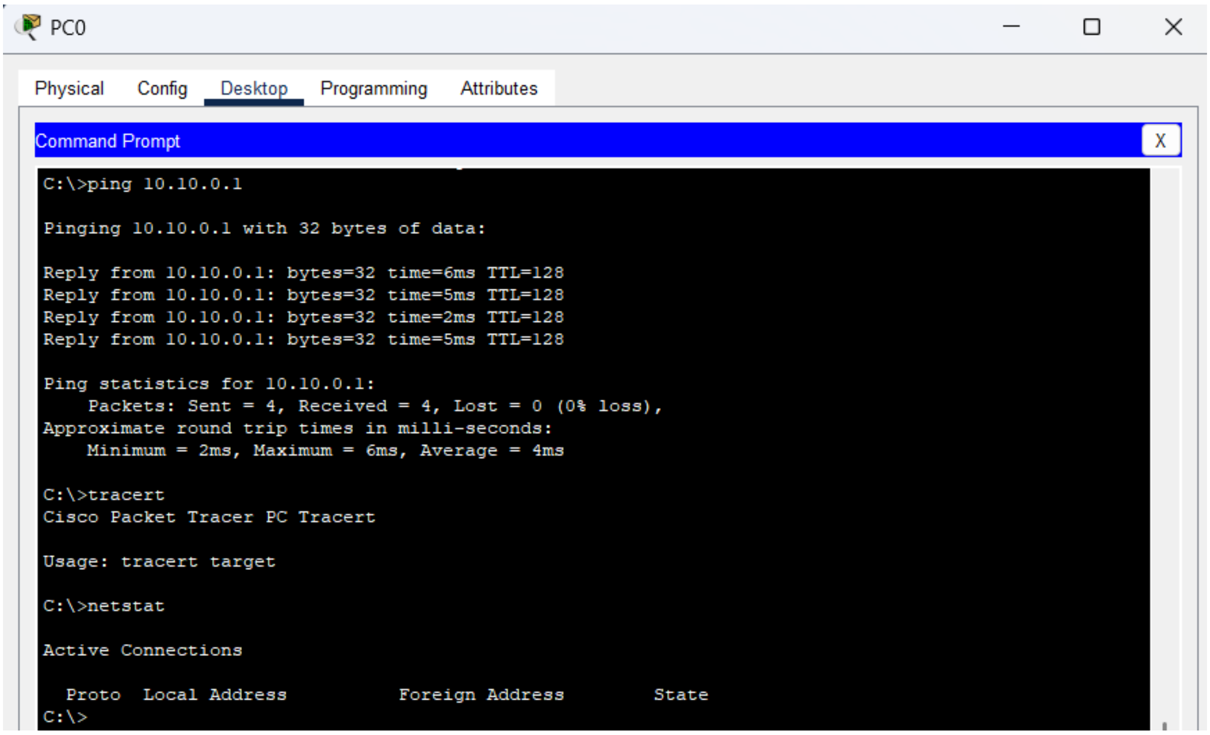
Step 3 :

- Assign IP (Internet Protocol) address to PC & Router.
- Go to Router0 > Config > Fast Ethernet 0/0 (interface you select) , Port status “ON” give IPv4 Address & Subnet Mask.(Example : IPv4 Address = 10.10.0.0 , Subnet Mask = 255.0.0.0)
- Go to PC-PT PC0 > Desktop > IP Configuration , give IPv4 Address & Subnet Mask.(Example : IPv4 Address = 10.10.0.1 , Subnet Mask = 255.0.0.0)
- Same for PC-PT PC1 .(Example : IPv4 Address = 10.10.0.2 , Subnet Mask = 255.0.0.0)



Step 4 :

- Click on PC > Desktop > Command Prompt.
- Run following command to check whether PC's are connected or not.
  - Ping 10.0.0.1
  - tracert[to trace route packet to destination]
  - netstat[to check connection & statistics.]



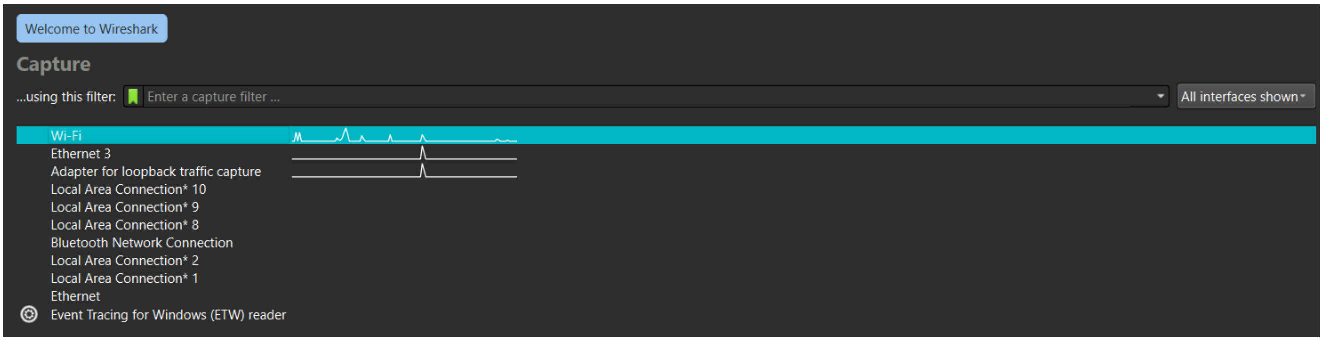
```
C:\>ping 10.10.0.1

Pinging 10.10.0.1 with 32 bytes of data:

Reply from 10.10.0.1: bytes=32 time=6ms TTL=128
Reply from 10.10.0.1: bytes=32 time=5ms TTL=128
Reply from 10.10.0.1: bytes=32 time=2ms TTL=128
Reply from 10.10.0.1: bytes=32 time=5ms TTL=128

Ping statistics for 10.10.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 6ms, Average = 4ms

C:\>tracert
Cisco Packet Tracer PC Tracert

Usage: tracert target

C:\>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
C:\>
```

PRACTICAL NO. 2

Aim : Perform Packet Sniffing

Step 1 :

- Open Wireshark
- Select you network connection



Welcome to Wireshark

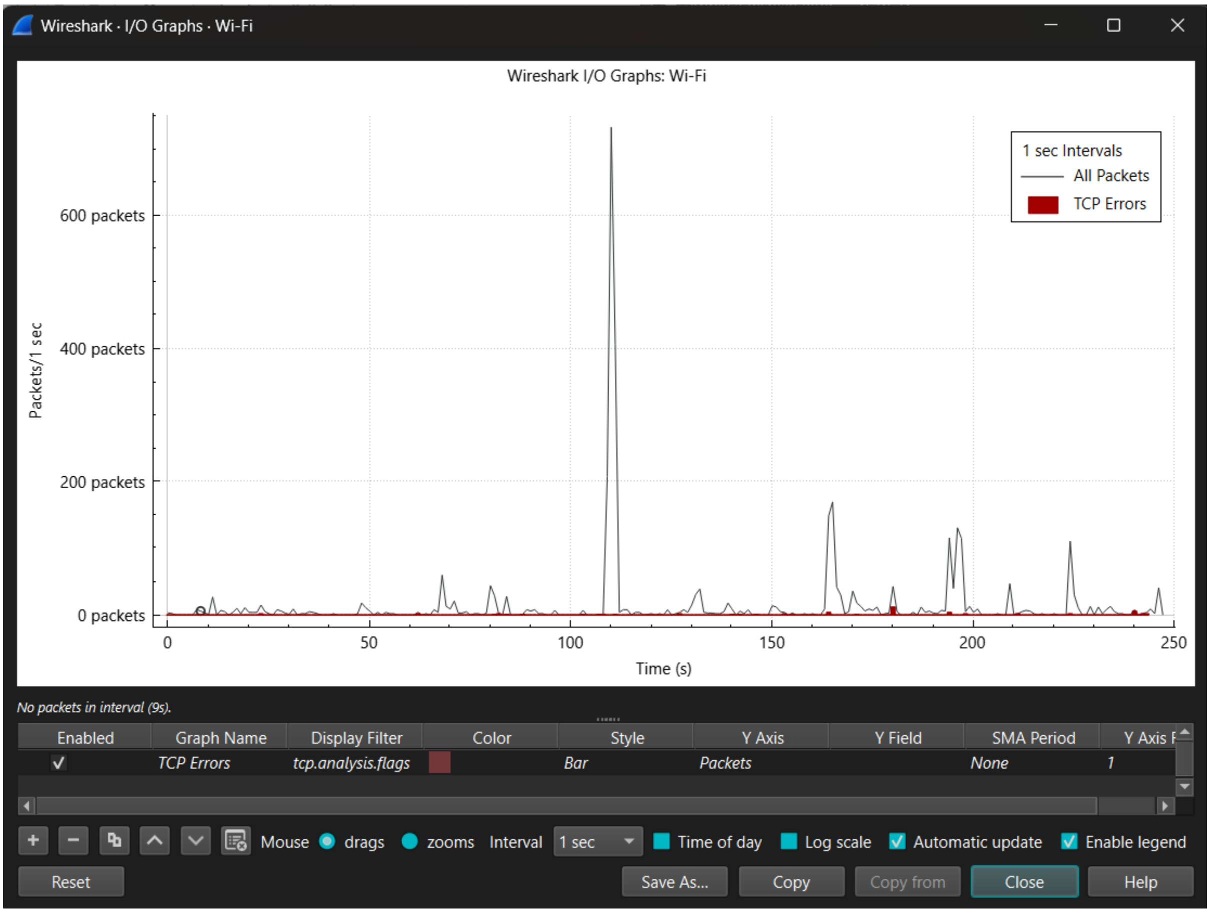
Capture

...using this filter: Enter a capture filter ... All interfaces shown

Wi-Fi
Ethernet 3
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 8
Bluetooth Network Connection
Local Area Connection* 2
Local Area Connection* 1
Ethernet
Event Tracing for Windows (ETW) reader

Step 2 :

- To see the I/O Graph
- In Navigation Bar Click on “Statictics” > “I/O Graph”



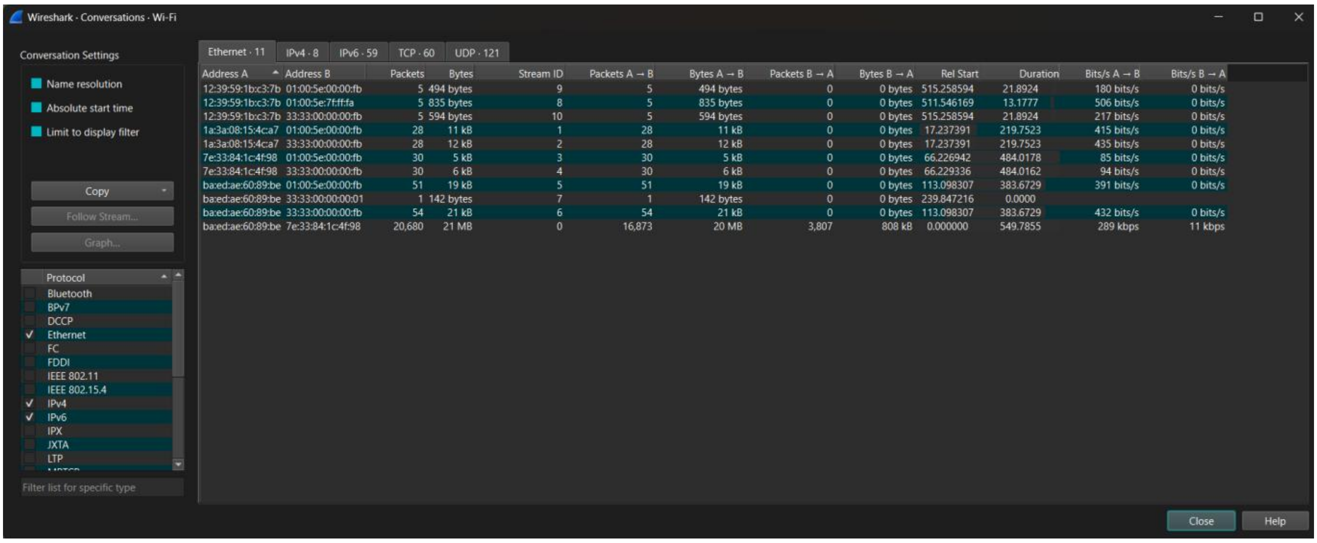
No packets in interval (9s).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis F
✓	TCP Errors	tcp.analysis.flags	🟥	Bar	Packets		None	1

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale ☑ Automatic update ☑ Enable legend

Step 3 :

- To check the conversion
- Go to the Navigation Bar Click on “Statictics” > “conversion”



Wireshark · Conversations · Wi-Fi

Conversation Settings

- Name resolution
- Absolute start time
- Limit to display filter

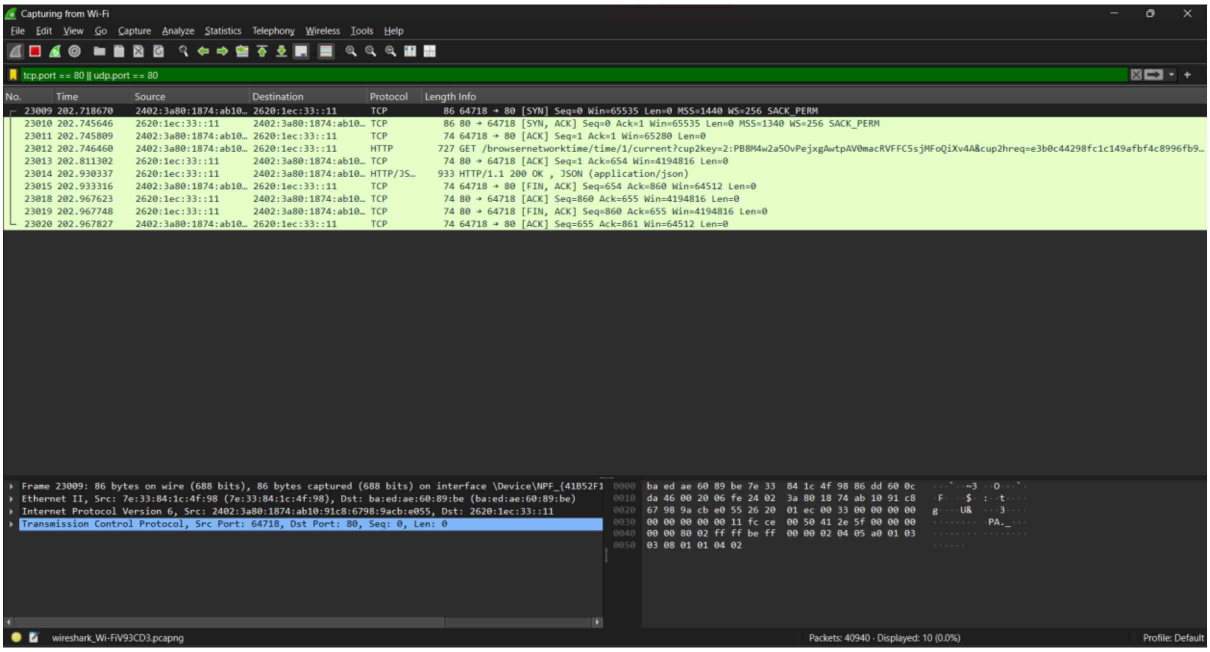
Protocol: Bluetooth, BPv7, DCCP, ✓ Ethernet, FC, FDDI, IEEE 802.11, IEEE 802.15.4, ✓ IPv4, ✓ IPv6, IPX, JXTA, LTP

Ethernet · 11 | IPv4 · 8 | IPv6 · 59 | TCP · 60 | UDP · 121

Address A	Address B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
12:39:59:1b:c3:7b	01:00:5e:00:00:fb	5	494 bytes	9	5	494 bytes	0	0 bytes	515.258594	21.8924	180 bits/s	0 bits/s
12:39:59:1b:c3:7b	01:00:5e:7f:ff:fa	5	835 bytes	8	5	835 bytes	0	0 bytes	511.546169	13.1777	506 bits/s	0 bits/s
12:39:59:1b:c3:7b	33:33:00:00:00:fb	5	594 bytes	10	5	594 bytes	0	0 bytes	515.258594	21.8924	217 bits/s	0 bits/s
1a:3a:08:15:4c:a7	01:00:5e:00:00:fb	28	11 kB	1	28	11 kB	0	0 bytes	17.237391	219.7523	415 bits/s	0 bits/s
1a:3a:08:15:4c:a7	33:33:00:00:00:fb	28	12 kB	2	28	12 kB	0	0 bytes	17.237391	219.7523	435 bits/s	0 bits/s
7e:33:84:1c:4f:98	01:00:5e:00:00:fb	30	5 kB	3	30	5 kB	0	0 bytes	66.226942	484.0178	85 bits/s	0 bits/s
7e:33:84:1c:4f:98	33:33:00:00:00:fb	30	6 kB	4	30	6 kB	0	0 bytes	66.229336	484.0162	94 bits/s	0 bits/s
ba:ed:ae:60:89:be	01:00:5e:00:00:fb	51	19 kB	5	51	19 kB	0	0 bytes	113.098307	383.6729	391 bits/s	0 bits/s
ba:ed:ae:60:89:be	33:33:00:00:00:01	1	142 bytes	7	1	142 bytes	0	0 bytes	239.847216	0.0000		
ba:ed:ae:60:89:be	33:33:00:00:00:fb	54	21 kB	6	54	21 kB	0	0 bytes	113.098307	383.6729	432 bits/s	0 bits/s
ba:ed:ae:60:89:be	7e:33:84:1c:4f:98	20,680	21 MB	0	16,873	20 MB	3,807	808 kB	0.000000	549.7855	289 kbps	11 kbps

Step 4 :

- Below the Navigation Bar “Apply a sisplay filter … <ctrl-/>”
- Write the follow line to add filter
    - tcp.port == 80 | | udp.port == 80



Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 80 || udp.port == 80

```
No.    Time        Source                 Destination            Protocol  Length Info
23009 202.718670 2402:3a80:1874:ab10… 2620:1ec:33::11        TCP        86 64718 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
23010 202.745646 2620:1ec:33::11      2402:3a80:1874:ab10…   TCP        86 80 → 64718 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1340 WS=256 SACK_PERM
23011 202.745809 2402:3a80:1874:ab10… 2620:1ec:33::11        TCP        74 64718 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
23012 202.746460 2402:3a80:1874:ab10… 2620:1ec:33::11        HTTP      727 GET /browsernetworktime/time/1/current?cup2key=2:PB8M4w2a5OvPejxgAwtpAV0macRVFFC5sjMFoQiXv4A&cup2hreq=e3b0c44298fc1c149afbf4c8996fb9…
23013 202.811302 2620:1ec:33::11      2402:3a80:1874:ab10…   TCP        74 80 → 64718 [ACK] Seq=1 Ack=654 Win=4194816 Len=0
23014 202.930337 2620:1ec:33::11      2402:3a80:1874:ab10…   HTTP/JS…  933 HTTP/1.1 200 OK , JSON (application/json)
23015 202.933316 2402:3a80:1874:ab10… 2620:1ec:33::11        TCP        74 64718 → 80 [FIN, ACK] Seq=654 Ack=860 Win=64512 Len=0
23018 202.967623 2620:1ec:33::11      2402:3a80:1874:ab10…   TCP        74 80 → 64718 [ACK] Seq=860 Ack=655 Win=4194816 Len=0
23019 202.967748 2620:1ec:33::11      2402:3a80:1874:ab10…   TCP        74 80 → 64718 [FIN, ACK] Seq=860 Ack=655 Win=4194816 Len=0
23020 202.967827 2402:3a80:1874:ab10… 2620:1ec:33::11        TCP        74 64718 → 80 [ACK] Seq=655 Ack=861 Win=64512 Len=0
```

```
Frame 23009: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{41B52F1
Ethernet II, Src: 7e:33:84:1c:4f:98 (7e:33:84:1c:4f:98), Dst: ba:ed:ae:60:89:be (ba:ed:ae:60:89:be)
Internet Protocol Version 6, Src: 2402:3a80:1874:ab10:91c8:6798:9acb:e055, Dst: 2620:1ec:33::11
Transmission Control Protocol, Src Port: 64718, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  ba ed ae 60 89 be 7e 33  84 1c 4f 98 86 dd 60 0c   ···`··~3 ··O···`·
0010  da 46 00 20 06 fe 24 02  3a 80 18 74 ab 10 91 c8   ·F· ··$· :··t····
0020  67 98 9a cb e0 55 26 20  01 ec 00 33 00 00 00 00   g····U&  ···3····
0030  00 00 00 00 00 11 fc ce  00 50 41 2e 5f 00 00 00   ········ ·PA._···
0040  00 00 80 02 ff ff be ff  00 00 02 04 05 a0 01 03   ········ ········
0050  03 08 01 01 04 02                                  ······
```

wireshark_Wi-FiV93CD3.pcapng Packets: 40940 · Displayed: 10 (0.0%) Profile: Default

PRACTICAL NO. 3

Aim : Integrating Wireshark with Nmap

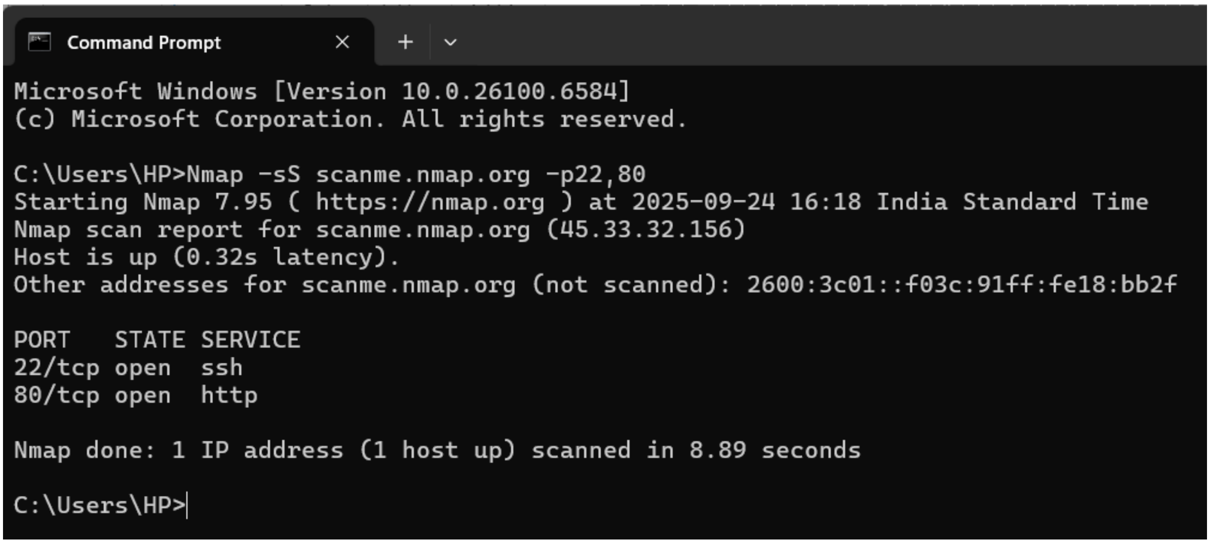
Nmap :- Nmap, short for Network Mapper, is a free and open-source tool used for network discovery and security auditing. It's primarily used to identify hosts, services, and open ports on a network, but it also offers a range of advanced features for network exploration and security assessments.

Step 1 : Perform Nmap Command

- Command :

  Nmap -sS scanme.nmap.org -p22,80

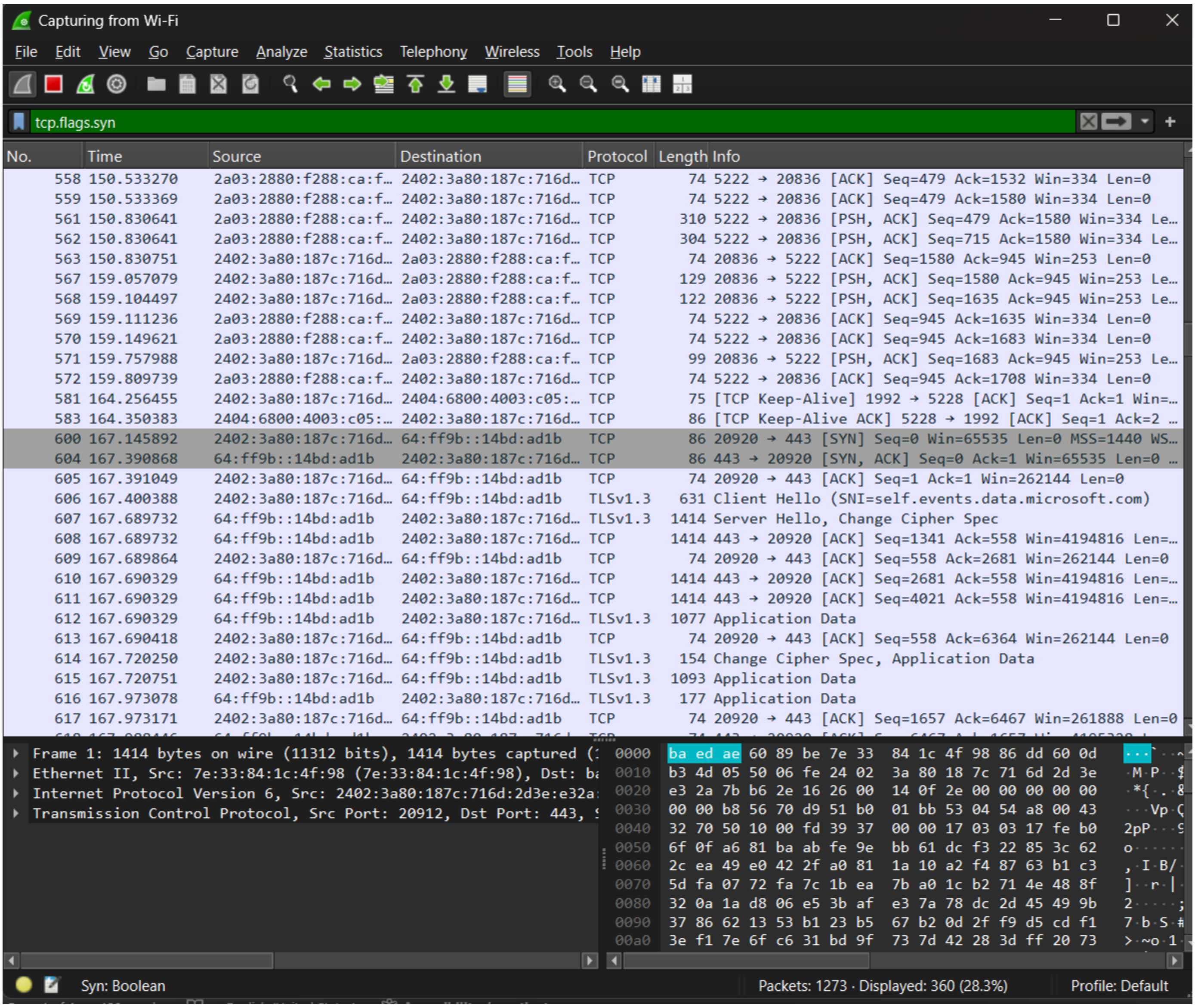
- It sends SYN packets to the target scanme.nmap.org on ports 22 and 80.
- Based on the response, it will tell you whether those ports are open or closed.
- It uses the SYN scan technique, which is faster and stealthier compared to a regular TCP connect scan.



```
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>Nmap -sS scanme.nmap.org -p22,80
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-24 16:18 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.32s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 8.89 seconds

C:\Users\HP>
```

Step 2 : Open Wire Shark and peform command in the filter

- tcp.flags.syn


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.syn

No.	Time	Source	Destination	Protocol	Length	Info
558	150.533270	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	74	5222 → 20836 [ACK] Seq=479 Ack=1532 Win=334 Len=0
559	150.533369	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	74	5222 → 20836 [ACK] Seq=479 Ack=1580 Win=334 Len=0
561	150.830641	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	310	5222 → 20836 [PSH, ACK] Seq=479 Ack=1580 Win=334 Le…
562	150.830641	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	304	5222 → 20836 [PSH, ACK] Seq=715 Ack=1580 Win=334 Le…
563	150.830751	2402:3a80:187c:716d…	2a03:2880:f288:ca:f…	TCP	74	20836 → 5222 [ACK] Seq=1580 Ack=945 Win=253 Len=0
567	159.057079	2402:3a80:187c:716d…	2a03:2880:f288:ca:f…	TCP	129	20836 → 5222 [PSH, ACK] Seq=1580 Ack=945 Win=253 Le…
568	159.104497	2402:3a80:187c:716d…	2a03:2880:f288:ca:f…	TCP	122	20836 → 5222 [PSH, ACK] Seq=1635 Ack=945 Win=253 Le…
569	159.111236	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	74	5222 → 20836 [ACK] Seq=945 Ack=1635 Win=334 Len=0
570	159.149621	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	74	5222 → 20836 [ACK] Seq=945 Ack=1683 Win=334 Len=0
571	159.757988	2402:3a80:187c:716d…	2a03:2880:f288:ca:f…	TCP	99	20836 → 5222 [PSH, ACK] Seq=1683 Ack=945 Win=253 Le…
572	159.809739	2a03:2880:f288:ca:f…	2402:3a80:187c:716d…	TCP	74	5222 → 20836 [ACK] Seq=945 Ack=1708 Win=334 Len=0
581	164.256455	2402:3a80:187c:716d…	2404:6800:4003:c05:…	TCP	75	[TCP Keep-Alive] 1992 → 5228 [ACK] Seq=1 Ack=1 Win=…
583	164.350383	2404:6800:4003:c05:…	2402:3a80:187c:716d…	TCP	86	[TCP Keep-Alive ACK] 5228 → 1992 [ACK] Seq=1 Ack=2 …
600	167.145892	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TCP	86	20920 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS…
604	167.390868	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TCP	86	443 → 20920 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 …
605	167.391049	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TCP	74	20920 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
606	167.400388	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TLSv1.3	631	Client Hello (SNI=self.events.data.microsoft.com)
607	167.689732	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TLSv1.3	1414	Server Hello, Change Cipher Spec
608	167.689732	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TCP	1414	443 → 20920 [ACK] Seq=1341 Ack=558 Win=4194816 Len=…
609	167.689864	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TCP	74	20920 → 443 [ACK] Seq=558 Ack=2681 Win=262144 Len=0
610	167.690329	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TCP	1414	443 → 20920 [ACK] Seq=2681 Ack=558 Win=4194816 Len=…
611	167.690329	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TCP	1414	443 → 20920 [ACK] Seq=4021 Ack=558 Win=4194816 Len=…
612	167.690329	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TLSv1.3	1077	Application Data
613	167.690418	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TCP	74	20920 → 443 [ACK] Seq=558 Ack=6364 Win=262144 Len=0
614	167.720250	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TLSv1.3	154	Change Cipher Spec, Application Data
615	167.720751	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TLSv1.3	1093	Application Data
616	167.973078	64:ff9b::14bd:ad1b	2402:3a80:187c:716d…	TLSv1.3	177	Application Data
617	167.973171	2402:3a80:187c:716d…	64:ff9b::14bd:ad1b	TCP	74	20920 → 443 [ACK] Seq=1657 Ack=6467 Win=261888 Len=0

```
▸ Frame 1: 1414 bytes on wire (11312 bits), 1414 bytes captured (
▸ Ethernet II, Src: 7e:33:84:1c:4f:98 (7e:33:84:1c:4f:98), Dst: ba
▸ Internet Protocol Version 6, Src: 2402:3a80:187c:716d:2d3e:e32a
▸ Transmission Control Protocol, Src Port: 20912, Dst Port: 443,
```

```
0000  ba ed ae 60 89 be 7e 33  84 1c 4f 98 86 dd 60 0d   ···`··~
0010  b3 4d 05 50 06 fe 24 02  3a 80 18 7c 71 6d 2d 3e   ·M·P··$
0020  e3 2a 7b b6 2e 16 26 00  14 0f 2e 00 00 00 00 00   ·*{·.·&
0030  00 00 b8 56 70 d9 51 b0  01 bb 53 04 54 a8 00 43   ···Vp·Q
0040  32 70 50 10 00 fd 39 37  00 00 17 03 03 17 fe b0   2pP···9
0050  6f 0f a6 81 ba ab fe 9e  bb 61 dc f3 22 85 3c 62   o······
0060  2c ea 49 e0 42 2f a0 81  1a 10 a2 f4 87 63 b1 c3   ,·I·B/·
0070  5d fa 07 72 fa 7c 1b ea  7b a0 1c b2 71 4e 48 8f   ]··r·|·
0080  32 0a 1a d8 06 e5 3b af  e3 7a 78 dc 2d 45 49 9b   2·····;
0090  37 86 62 13 53 b1 23 b5  67 b2 0d 2f f9 d5 cd f1   7·b·S·#
00a0  3e f1 7e 6f c6 31 bd 9f  73 7d 42 28 3d ff 20 73   >·~o·1·
```

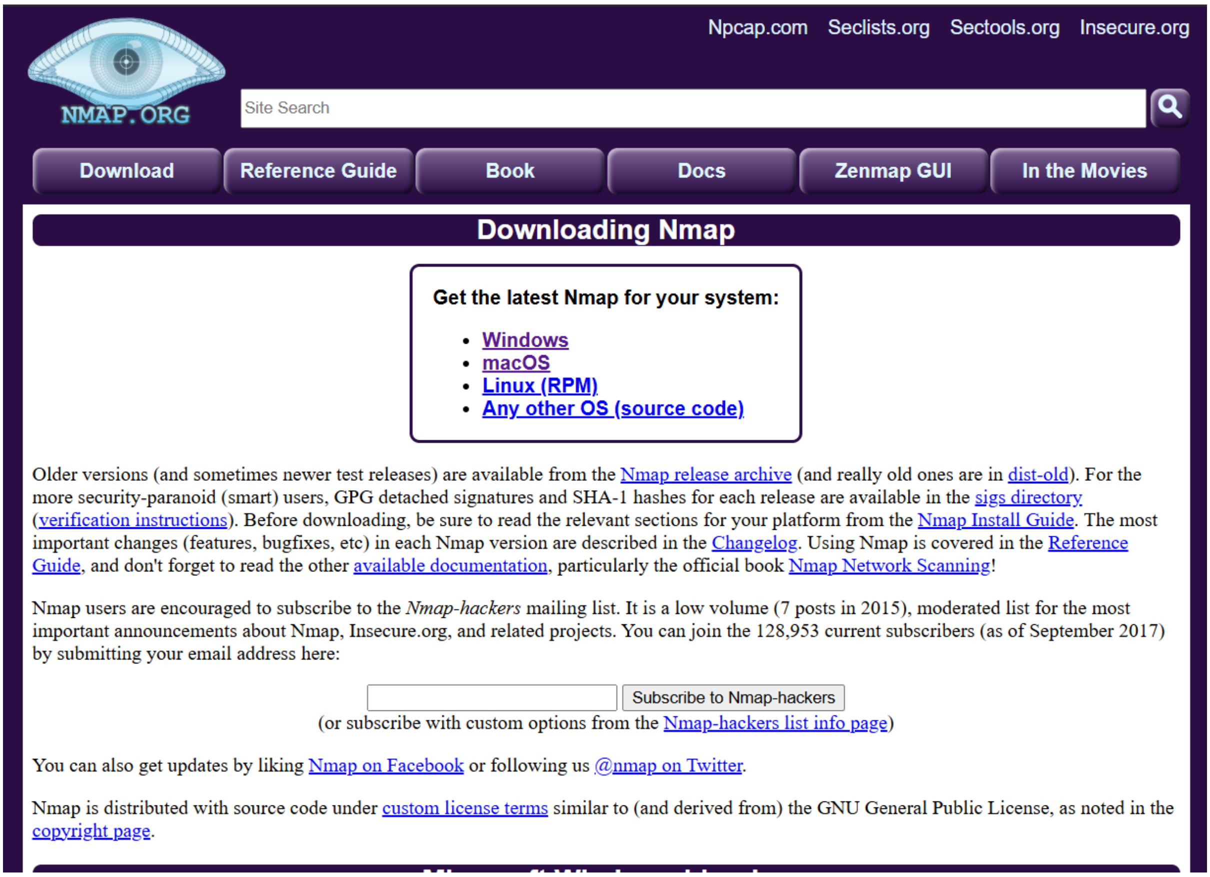
Syn: Boolean Packets: 1273 · Displayed: 360 (28.3%) Profile: Default

PRACTICAL NO. 4

Aim : Perform Port Scanning to find out open and closed ports over the network

Step 1 : Download Zenmap

- Download Zenmap (Nmap) network scanning tool.
- Search on browser "zenmap download" or go to the official website https://nmap.org/zenmap
- Select your system and download.



Downloading Nmap

Get the latest Nmap for your system:

- Windows
- macOS
- Linux (RPM)
- Any other OS (source code)

Older versions (and sometimes newer test releases) are available from the Nmap release archive (and really old ones are in dist-old). For the more security-paranoid (smart) users, GPG detached signatures and SHA-1 hashes for each release are available in the sigs directory (verification instructions). Before downloading, be sure to read the relevant sections for your platform from the Nmap Install Guide. The most important changes (features, bugfixes, etc) in each Nmap version are described in the Changelog. Using Nmap is covered in the Reference Guide, and don't forget to read the other available documentation, particularly the official book Nmap Network Scanning!

Nmap users are encouraged to subscribe to the *Nmap-hackers* mailing list. It is a low volume (7 posts in 2015), moderated list for the most important announcements about Nmap, Insecure.org, and related projects. You can join the 128,953 current subscribers (as of September 2017) by submitting your email address here:

Subscribe to Nmap-hackers
(or subscribe with custom options from the Nmap-hackers list info page)

You can also get updates by liking Nmap on Facebook or following us @nmap on Twitter.

Nmap is distributed with source code under custom license terms similar to (and derived from) the GNU General Public License, as noted in the copyright page.

Step 2 : Installation

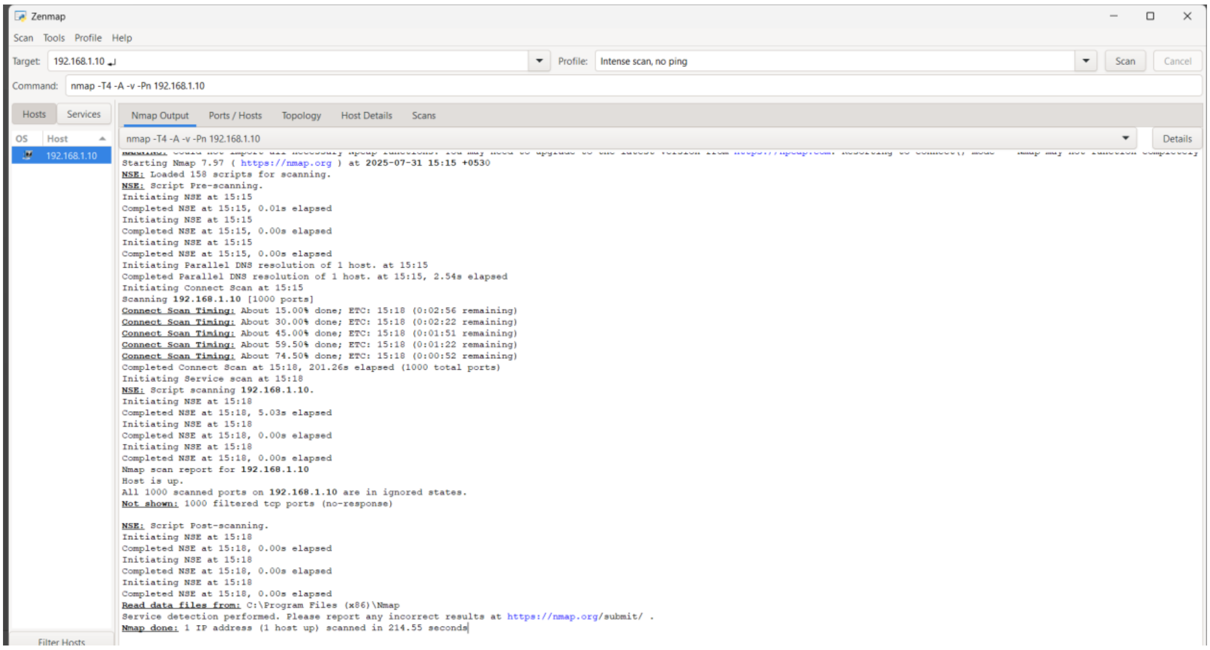
- Agree the Term & Conditions and install it.
- Launch Zenmap from your applications menu.

Step 3 : Perform different types of network scanning

1. Intense Scan, No Ping

- In the Target field, enter your target IP or domain.
- In the Profile dropdown, select: "Intense scan, no ping"
- You'll see this command auto-filled in the Command field: "nmap -T4 -A -v -Pn 192.168.1.10"

- Click the Scan button



Target: 192.168.1.10

Profile: Intense scan, no ping

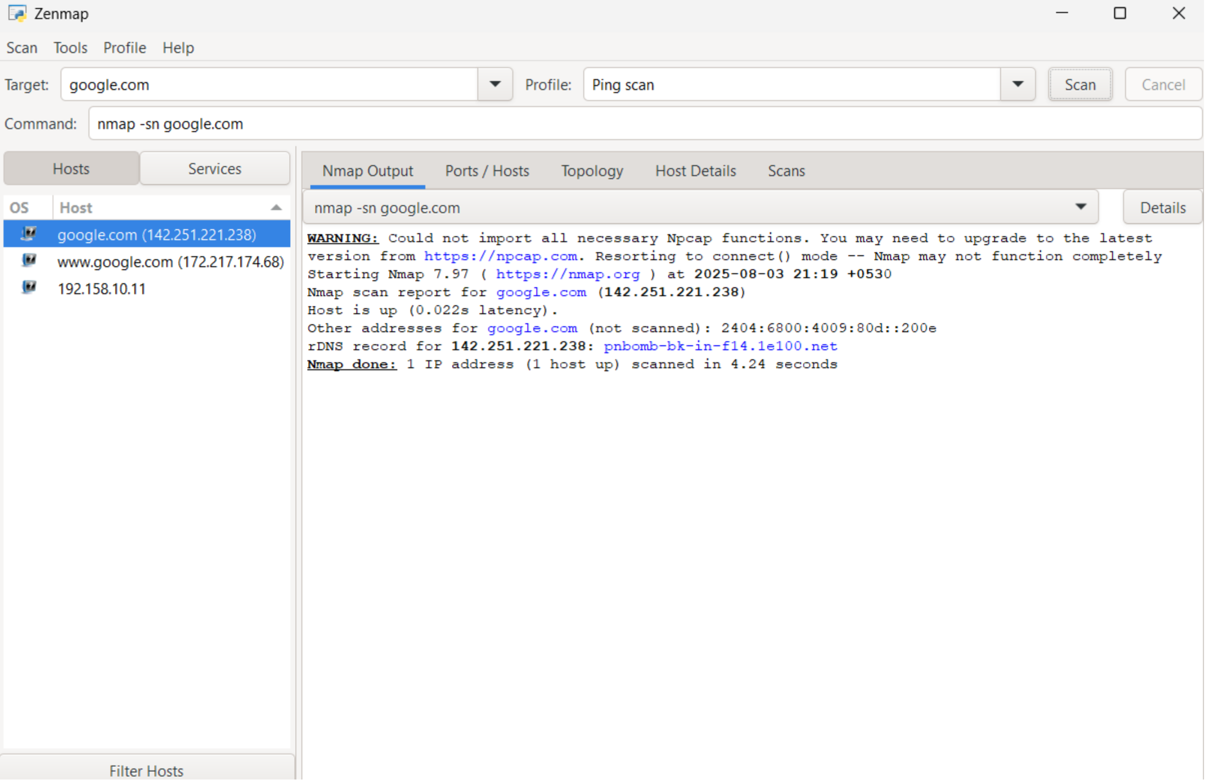
Command: nmap -T4 -A -v -Pn 192.168.1.10

Host: 192.168.1.10

```
nmap -T4 -A -v -Pn 192.168.1.10
WARNING: Could not import all necessary Npcap functions. You may need to upgrade to the latest version from https://npcap.com. Resorting to connect() mode -- Nmap may not function completely
Starting Nmap 7.97 ( https://nmap.org ) at 2025-07-31 15:15 +0530
NSE: Loaded 158 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:15
Completed NSE at 15:15, 0.01s elapsed
Initiating NSE at 15:15
Completed NSE at 15:15, 0.00s elapsed
Initiating NSE at 15:15
Completed NSE at 15:15, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 15:15
Completed Parallel DNS resolution of 1 host. at 15:15, 2.54s elapsed
Initiating Connect Scan at 15:15
Scanning 192.168.1.10 [1000 ports]
Connect Scan Timing: About 15.00% done; ETC: 15:18 (0:02:56 remaining)
Connect Scan Timing: About 30.00% done; ETC: 15:18 (0:02:22 remaining)
Connect Scan Timing: About 45.00% done; ETC: 15:18 (0:01:51 remaining)
Connect Scan Timing: About 59.50% done; ETC: 15:18 (0:01:22 remaining)
Connect Scan Timing: About 74.50% done; ETC: 15:18 (0:00:52 remaining)
Completed Connect Scan at 15:18, 201.26s elapsed (1000 total ports)
Initiating Service scan at 15:18
NSE: Script scanning 192.168.1.10.
Initiating NSE at 15:18
Completed NSE at 15:18, 5.03s elapsed
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Nmap scan report for 192.168.1.10
Host is up.
All 1000 scanned ports on 192.168.1.10 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

NSE: Script Post-scanning.
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 214.55 seconds
```

2. Ping Scan (nmap -sn google.com)

- In the Target field, enter your target IP or domain and Profile select: “Ping scan”.



Target: google.com

Profile: Ping scan

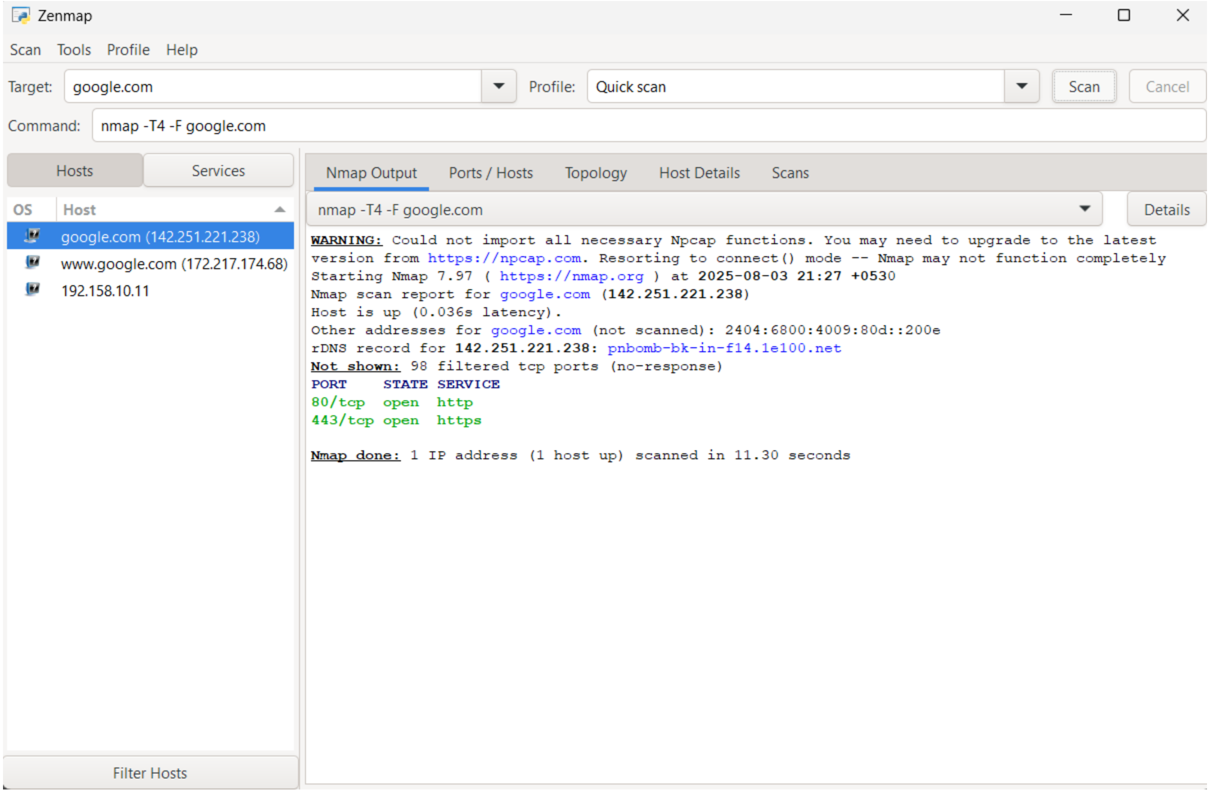
Command: nmap -sn google.com

Hosts:
google.com (142.251.221.238)
www.google.com (172.217.174.68)
192.158.10.11

```
nmap -sn google.com
WARNING: Could not import all necessary Npcap functions. You may need to upgrade to the latest
version from https://npcap.com. Resorting to connect() mode -- Nmap may not function completely
Starting Nmap 7.97 ( https://nmap.org ) at 2025-08-03 21:19 +0530
Nmap scan report for google.com (142.251.221.238)
Host is up (0.022s latency).
Other addresses for google.com (not scanned): 2404:6800:4009:80d::200e
rDNS record for 142.251.221.238: pnbomb-bk-in-f14.1e100.net
Nmap done: 1 IP address (1 host up) scanned in 4.24 seconds
```

3. Quick Scan (nmap -T4 -F google.com)

- In the Target field, enter your target IP or domain and Profile select: “Quick Scan”.



Target: google.com

Profile: Quick scan

Command: nmap -T4 -F google.com

google.com (142.251.221.238)
www.google.com (172.217.174.68)
192.158.10.11

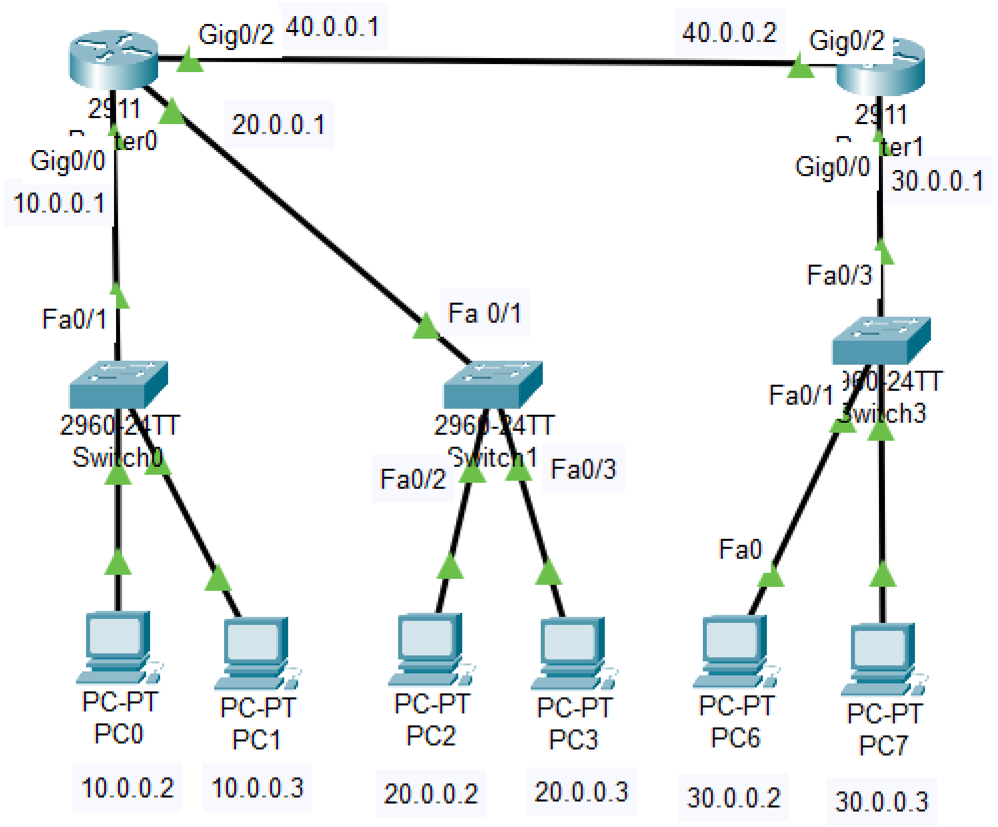
nmap -T4 -F google.com

```
WARNING: Could not import all necessary Npcap functions. You may need to upgrade to the latest
version from https://npcap.com. Resorting to connect() mode -- Nmap may not function completely
Starting Nmap 7.97 ( https://nmap.org ) at 2025-08-03 21:27 +0530
Nmap scan report for google.com (142.251.221.238)
Host is up (0.036s latency).
Other addresses for google.com (not scanned): 2404:6800:4009:80d::200e
rDNS record for 142.251.221.238: pnbomb-bk-in-f14.1e100.net
Not shown: 98 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 11.30 seconds
```

PRACTICAL NO. 5

Aim : Perform ACL (Access Control List)

Step 1 : Topology

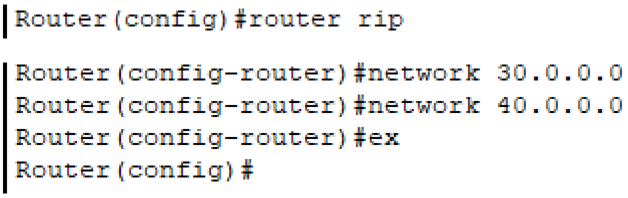


Step 2 :

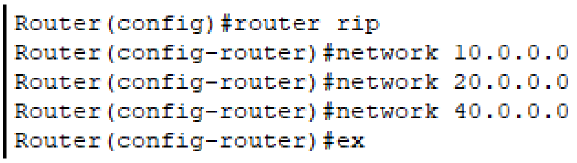
- First do all the basic configuration and provide IP address to all devices like Pc & router.

Step 3 :

- In order to ping each other first do RIP protocol in every router with the following commands.



```
Router(config)#router rip

Router(config-router)#network 30.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#ex
Router(config)#
```



```
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#ex
```

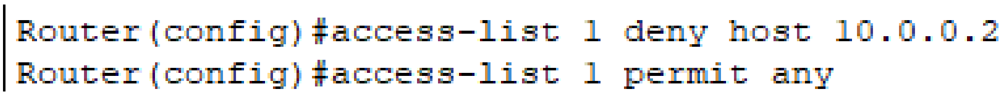
Note : type the IP that you have given in the network.

Step 4 :

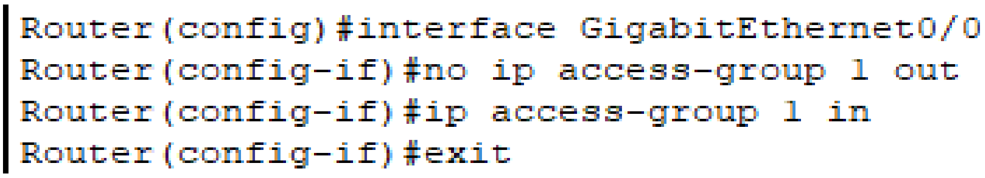
- Now ping the Pc first and check whether topology is perfectly working or not.

Step 5 :

- Do main ACL command for standard one-one method in the source router i.e. the first router in the network R1.



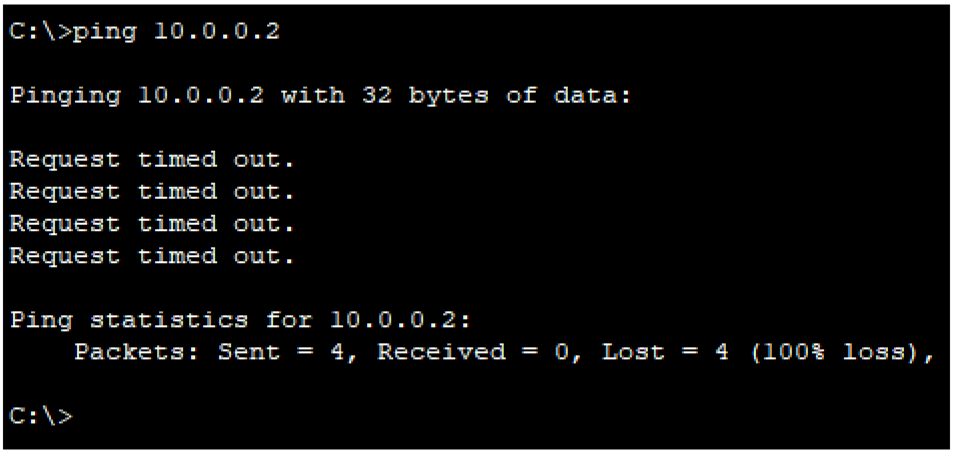
```
Router(config)#access-list 1 deny host 10.0.0.2
Router(config)#access-list 1 permit any
```



```
Router(config)#interface GigabitEthernet0/0
Router(config-if)#no ip access-group 1 out
Router(config-if)#ip access-group 1 in
Router(config-if)#exit
```

Step 6 :

- Now check the ping process for 10.0.0.2 with all Pc it should be blocking 10.0.0.2 host machine.



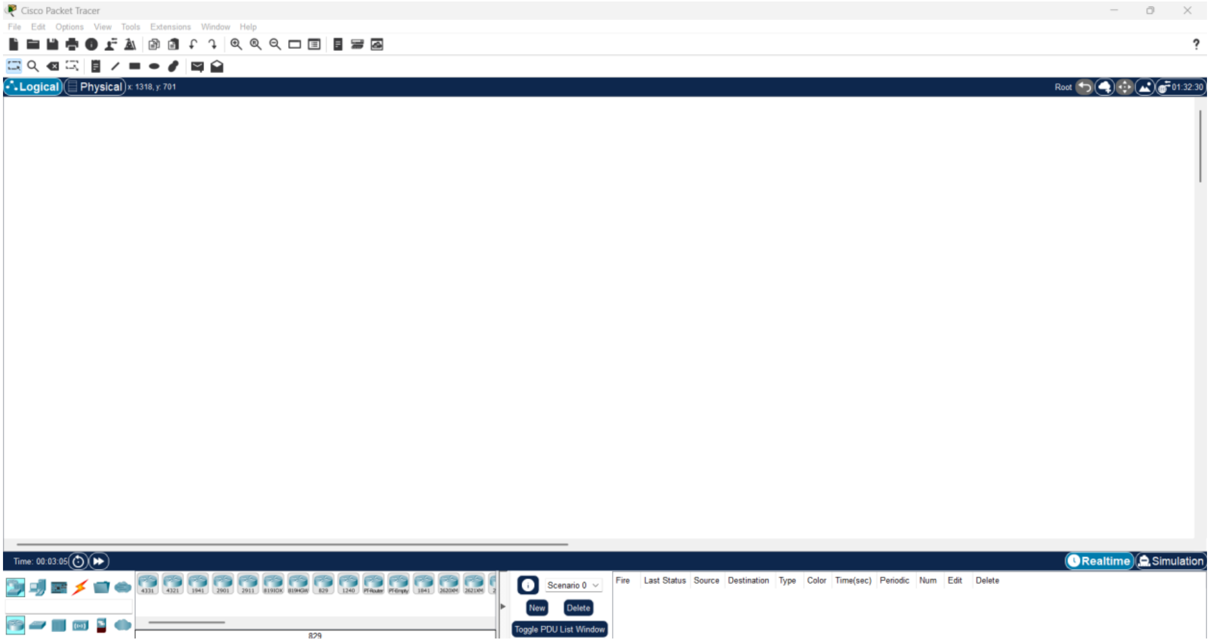
```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

PRACTICAL NO. 6

Aim : Configuring DHCP to show how dynamic IP given to the network

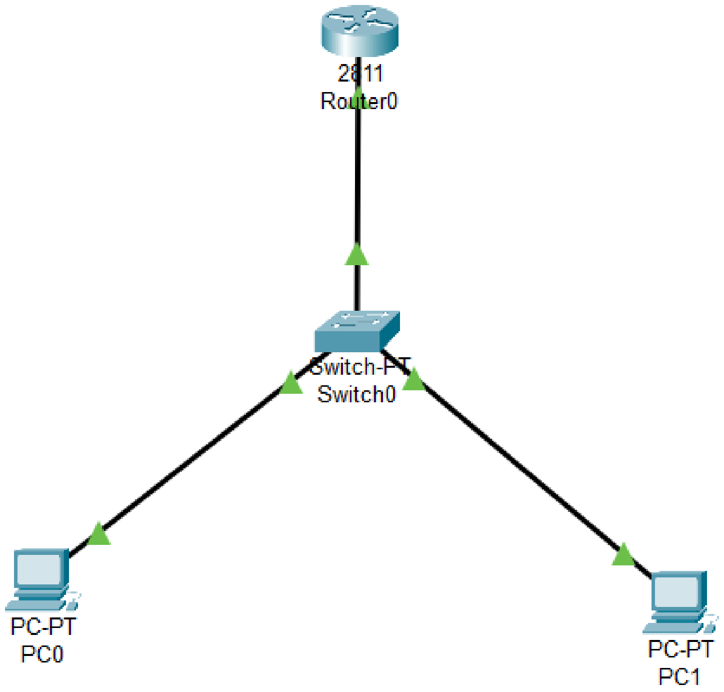
Step 1 :

- Open Cisco Packet Tracer



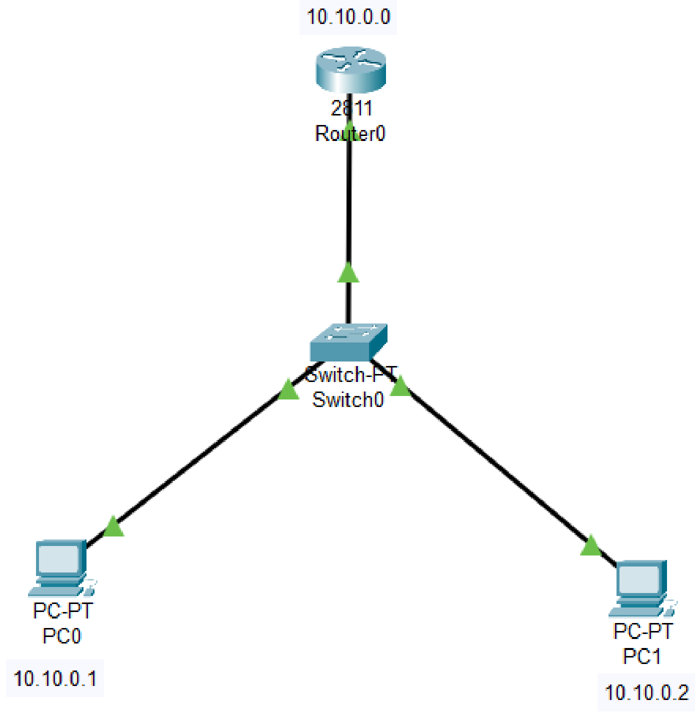
Step 2 :

- Take 2 PC, Switch and Router.
- Make appropriate connection by using "Copper Straight-Through".



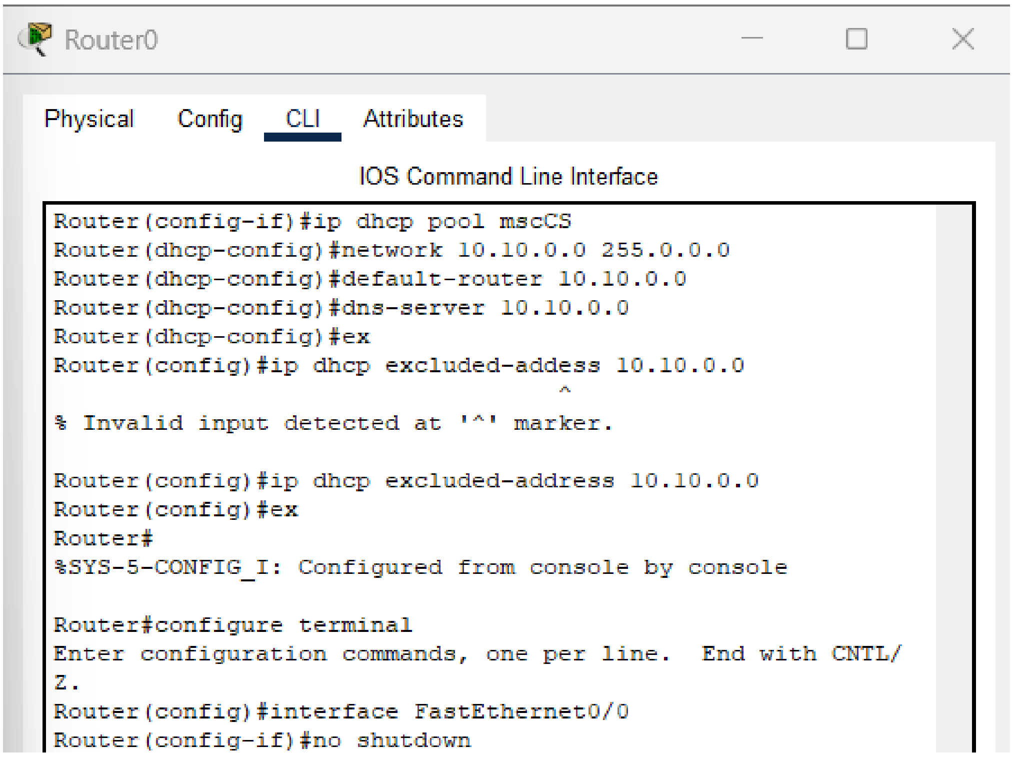
Step 3 :

- Assign IP (Internet Protocol) address to PC & Router.
- Go to Router0 > Config > Fast Ethernet 0/0 (interface you select) , Port status “ON” give IPv4 Address & Subnet Mask.(Example : IPv4 Address = 10.0.0.0 , Subnet Mask = 255.0.0.0)
- Go to PC-PT PC0 > Desktop > IP Configuration , give IPv4 Address & Subnet Mask.(Example : IPv4 Address = 10.0.0.1 , Subnet Mask = 255.0.0.0)
- Same for PC-PT PC1 .(Example : IPv4 Address = 10.0.0.2 , Subnet Mask = 255.0.0.0)



Step 4 :

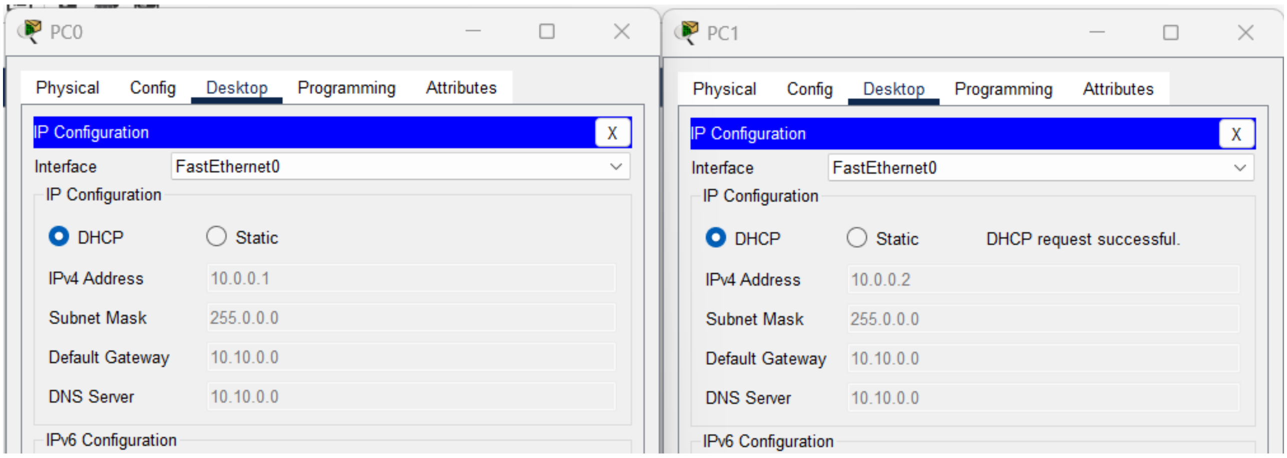
- Click on Router > CLI , Perform the DHCP Configuration commands as per below:
  - ip dhcp pool mscCS
  - network 10.10.0.0 255.0.0.0
  - default-router 10.10.0.0
  - ex
  - ip dhcp excluded-address 10.10.0.0



```
Router(config-if)#ip dhcp pool mscCS
Router(dhcp-config)#network 10.10.0.0 255.0.0.0
Router(dhcp-config)#default-router 10.10.0.0
Router(dhcp-config)#dns-server 10.10.0.0
Router(dhcp-config)#ex
Router(config)#ip dhcp excluded-address 10.10.0.0
                                       ^
% Invalid input detected at '^' marker.

Router(config)#ip dhcp excluded-address 10.10.0.0
Router(config)#ex
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/
Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown
```

Step 5 :

- Open the both the PCs go to desktop > Ip configuration ,Tick the DHCP option for request and assign ip address



Interface: FastEthernet0
IP Configuration
● DHCP ○ Static
IPv4 Address: 10.0.0.1
Subnet Mask: 255.0.0.0
Default Gateway: 10.10.0.0
DNS Server: 10.10.0.0

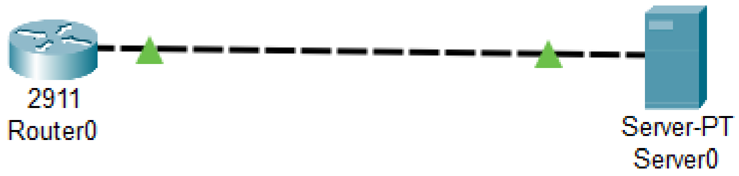
Interface: FastEthernet0
IP Configuration
● DHCP ○ Static DHCP request successful.
IPv4 Address: 10.0.0.2
Subnet Mask: 255.0.0.0
Default Gateway: 10.10.0.0
DNS Server: 10.10.0.0

PRACTICAL NO. 7

Aim : Perform Backup & Restore of Running Config.

Step 1 : Backup

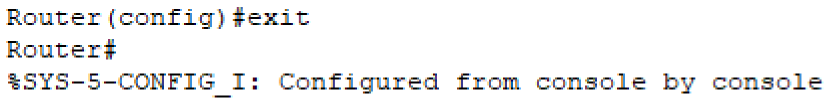
- Take router 2911 and server pt, connect it by using copper cross-over



- Assign IP to Router = 10.0.0.1

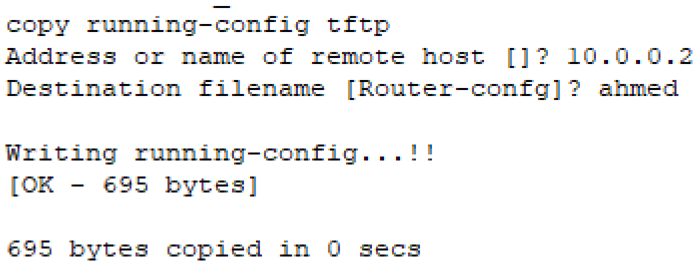
  Server = 10.0.0.2

- Open router "CLI" and if you in (config-if)/(config) then exit bu using command "exit"



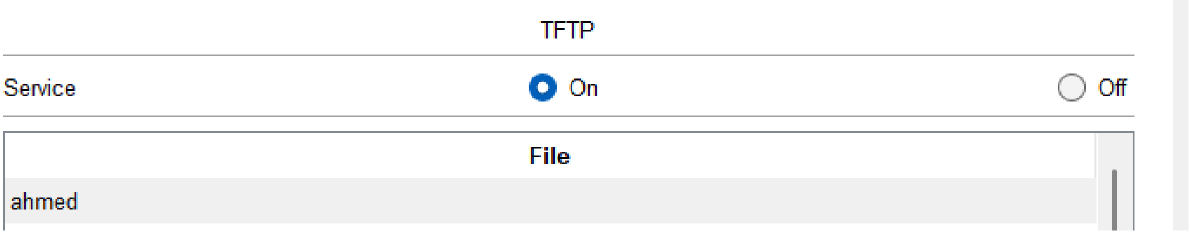
```
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

- Type command : copy running-config tftp
- Give the router ip 10.0.0.1
- Give file name (what you want) :ahmed



```
copy running-config tftp
Address or name of remote host []? 10.0.0.2
Destination filename [Router-confg]? ahmed

Writing running-config...!!
[OK - 695 bytes]

695 bytes copied in 0 secs
```

- Oper your server in services > tftp
- Here you can see the you file ahmed



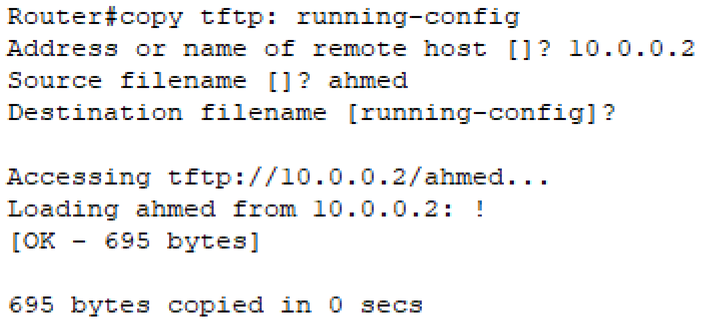
TFTP

Service ● On ○ Off

File
ahmed

Step 2 : Restore

- Same for restore
- Open router “CLI”
- Give command for restore : “copy tftp: running-config”
- Give your server IP 10.0.0.2, and file name ahmed, and press enter on destination filename



```
Router#copy tftp: running-config
Address or name of remote host []? 10.0.0.2
Source filename []? ahmed
Destination filename [running-config]?

Accessing tftp://10.0.0.2/ahmed...
Loading ahmed from 10.0.0.2: !
[OK - 695 bytes]

695 bytes copied in 0 secs
```

PRACTICAL NO. 8

Aim : Solve Sums on subnetting

1 . 192.168.10.11/24 where host is 50 , Find new network & New Broadcast.

Solution :

Step 1 :

Network = 192.168.10.11/24

- As it is belong to class C
- As host is 50 ∴ 50+2=52 hosts

Step 2 :

- Subnet Mark = 255.255.255.0

Step 3 :

Wild Card Mask (WCM) = 0.0.0.255

Step 4 :

- Choose from range suitable to host 52

Step 5 :

$2^6 = 64$ [0 - 63]

$n = 2^n$

$n = 6$

Step 6 :

- Total Network bit is 32
  = 32 – n
  = 32 – 6
  = 26

Step 7 :

11111111.11111111.11111111.11000000

Step 8 :

- New Network = 192.168.10.192/26
- New Broadcast = 192.168.10.255/26

2 . (92.(68.10.1)/24 where host is 28, Find new network & new broadcast.

Solution :

Step 1 :

Network = (92.(68.10.1)/24
As it belong to class 'C'
and host = 28 ∴ 28+2 = 30 hosts

Step 2 :

- Subnet Mask = 255.255.255.0

Step 3 :

Wild Card Mask (WCM) = 0.0.0.255

Step 4:

- Choose from range suitable to host 30

Step 5 :

$2^5 = 32$ [0-31]

$n = 2^n$

$n = 5$

Step 6 :

- The network bit is 32
  ∴ 32 – n
  = 32 – 5 = 27

Step 7 :

11111111.11111111.11111111.11100000

Step 8 :

- New Network = 192.168.10.224/27
- New Broadcast = 192.168.10.255/27