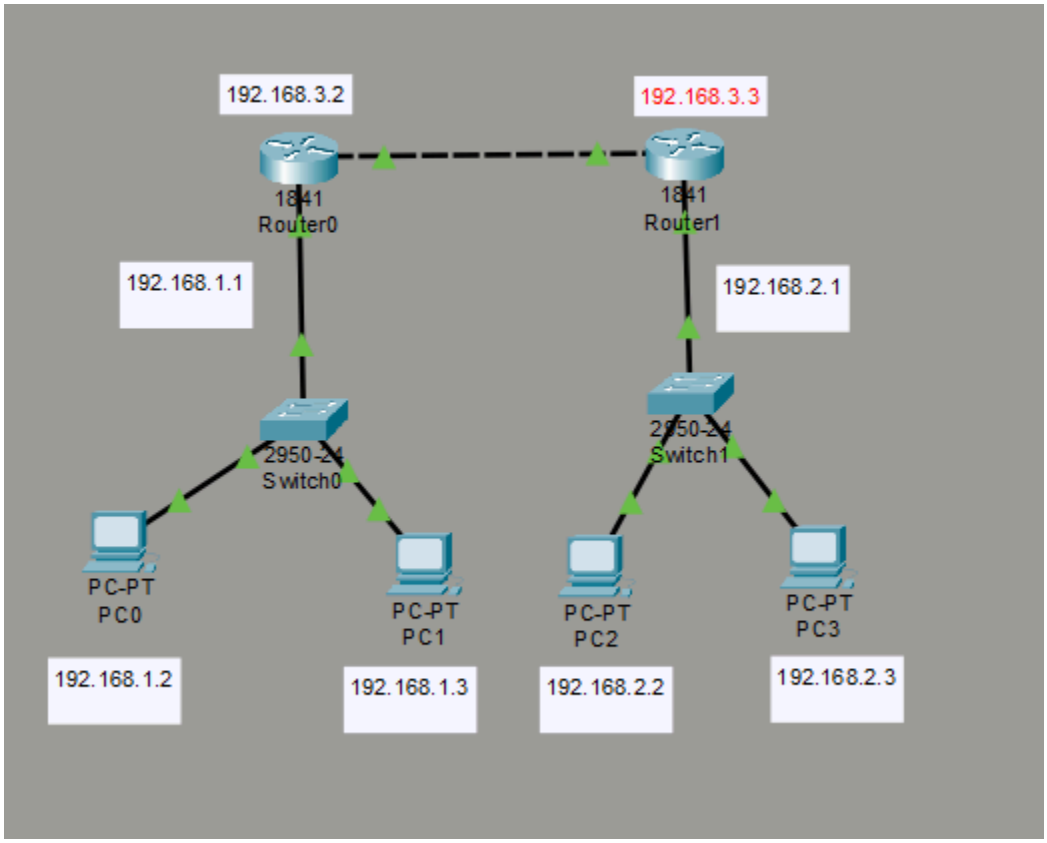
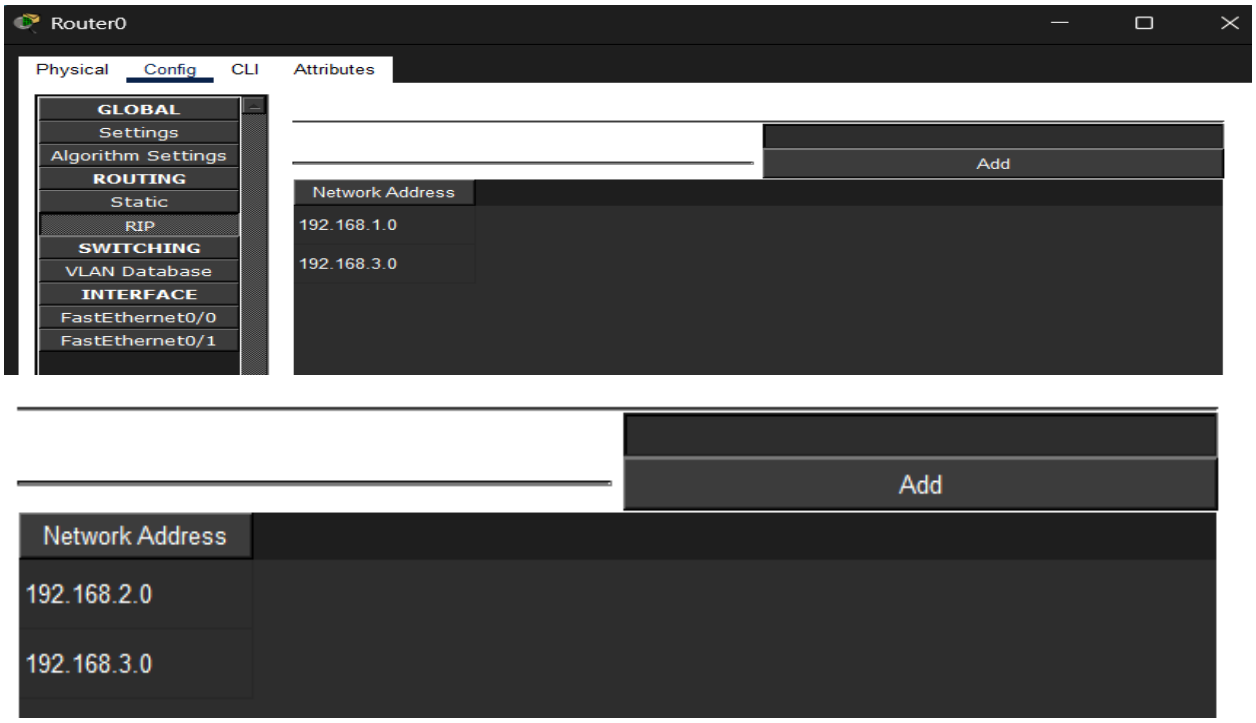
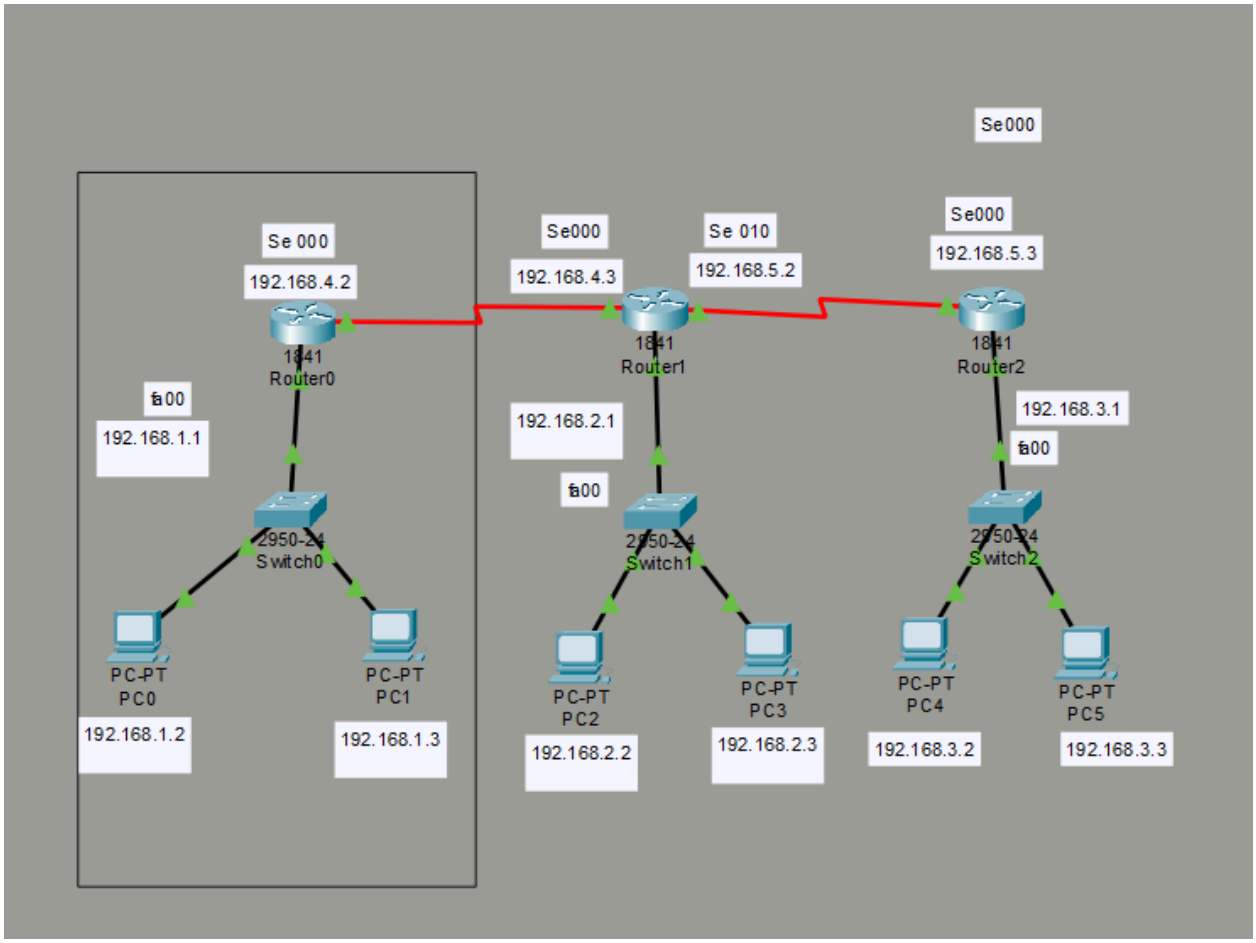
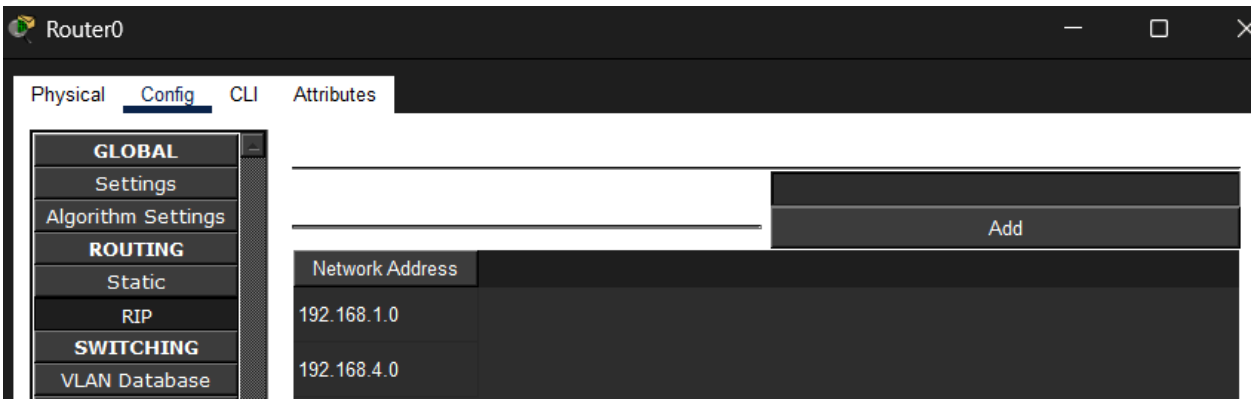
RIP FOR 2 ROUTER

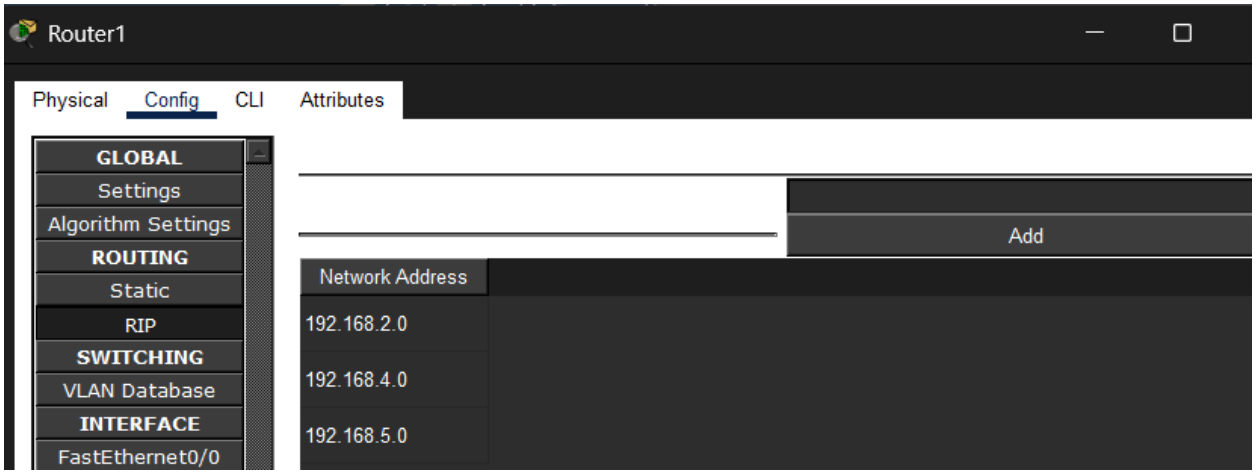


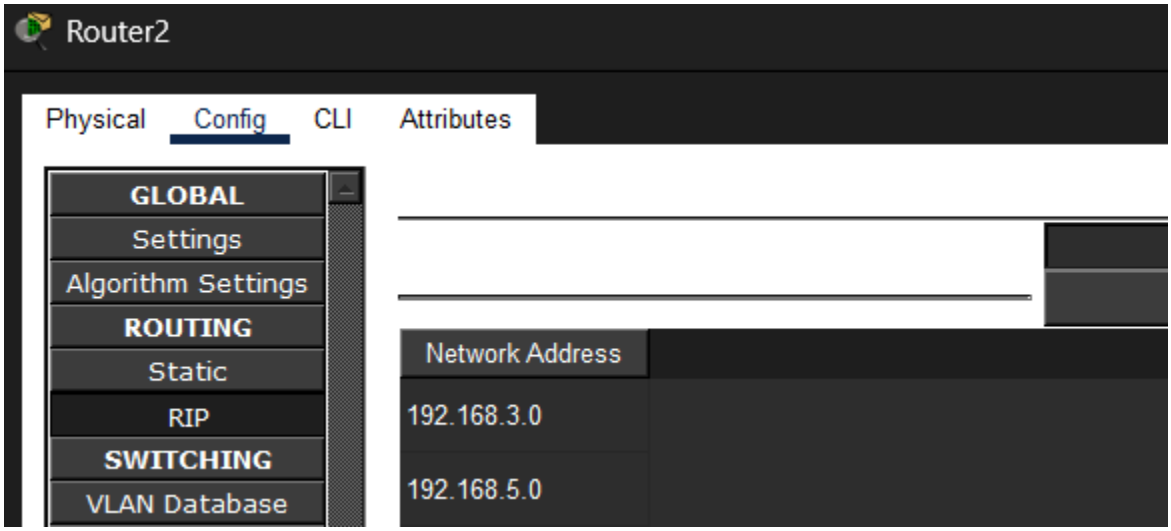


RIP USING 3 ROUTERS

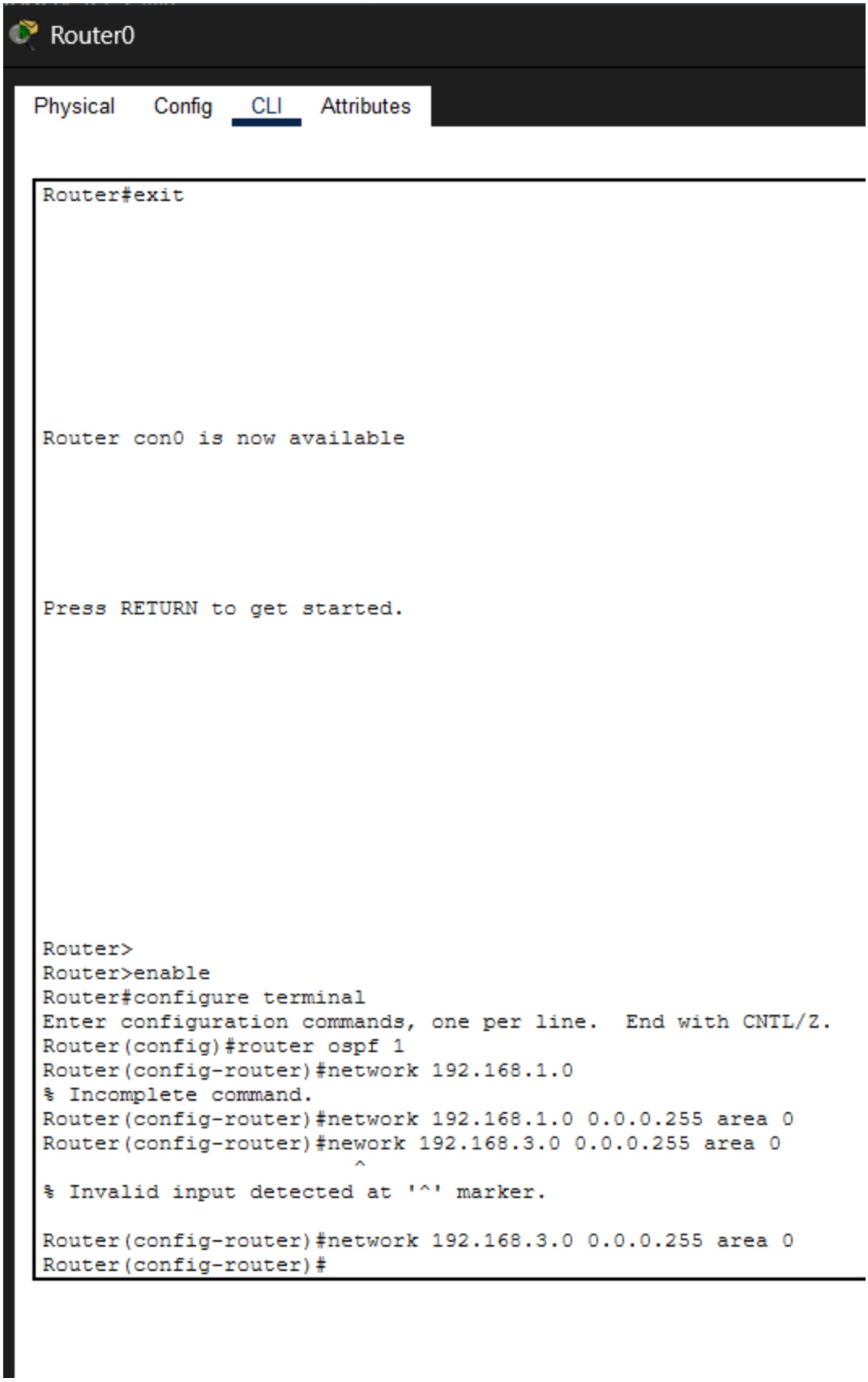





Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
Add
Network Address
192.168.2.0
192.168.4.0
192.168.5.0


Router2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
Network Address
192.168.3.0
192.168.5.0

OSPF WITH 2


Router0
Physical
Config
CLI
Attributes
Router#exit
Router con0 is now available
Press RETURN to get started.
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.1.0
% Incomplete command.
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router(config-router)#nework 192.168.3.0 0.0.0.255 area 0
^
% Invalid input detected at '^' marker.
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 2
Router(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#
Router(config-router)#exit
Router(config)#
```

router 1

Rest at amit saw folder

Practical 6

```
import ipaddress

def subnet_calculator():
    while True:
        print("\n--- IP Subnetting Calculator ---")
        print("1. Calculate subnet details")
        print("2. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            ip_input = input("Enter IP address with CIDR (e.g., 205.16.37.39/28): ")

            try:
                network = ipaddress.ip_network(ip_input, strict=False)

                print("\nResults:")
                print(f"Network ID       : {network.network_address}")
                print(f"Subnet Mask      : {network.netmask}")
                print(f"Broadcast Address: {network.broadcast_address}")

                all_hosts = list(network.hosts())
                print(f"First Usable     : {all_hosts[0] if all_hosts else 'N/A'}")
                print(f"Last Usable      : {all_hosts[-1] if all_hosts else 'N/A'}")
                print(f"Total Addresses  : {network.num_addresses}")
                print(f"Usable Hosts     : {len(all_hosts)}")

            except ValueError:
                print("Invalid IP address or CIDR notation. Try again.")
```

```
    elif choice == '2':
        print("Exiting program.")
        break
    else:
        print("Invalid choice. Try again.")

subnet_calculator()
```

Practical 8

```
import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) {
        try (ServerSocket serverSocket = new ServerSocket(6000)) {
            System.out.println("Server started. Waiting for client...");

            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected");

            // Receive data from client
            BufferedReader input = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
            String received = input.readLine();
            System.out.println("Received from client: " + received);

            input.close();
            clientSocket.close();

        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}


import java.io.*;
import java.net.*;
```

```java
public class Client {
    public static void main(String[] args) {
        try (Socket socket = new Socket("localhost", 6000)) {
            System.out.println("Connected to server.");

            // Send data to server
            PrintWriter output = new PrintWriter(socket.getOutputStream(),
true);
            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));

            System.out.print("Enter a message for the server: ");
            String message = userInput.readLine();
            output.println(message);
            System.out.println("Message sent to server.");

            output.close();
            userInput.close();

        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

javac Server.java
javac Client.java

Then in 1 —>terminal java Server
Then in 2nd —>terminal java Client