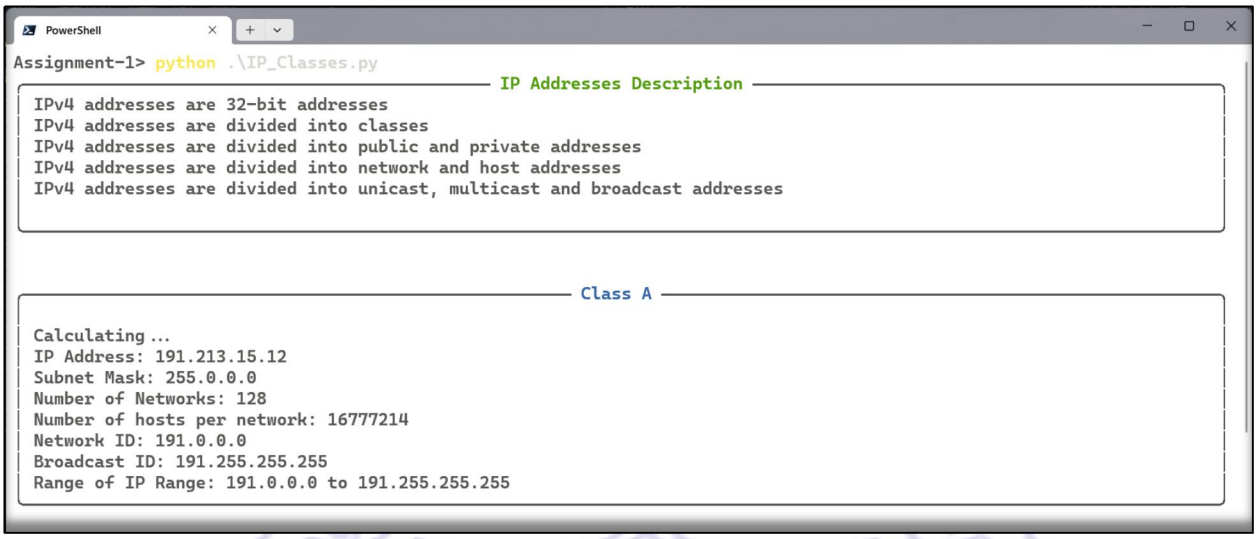
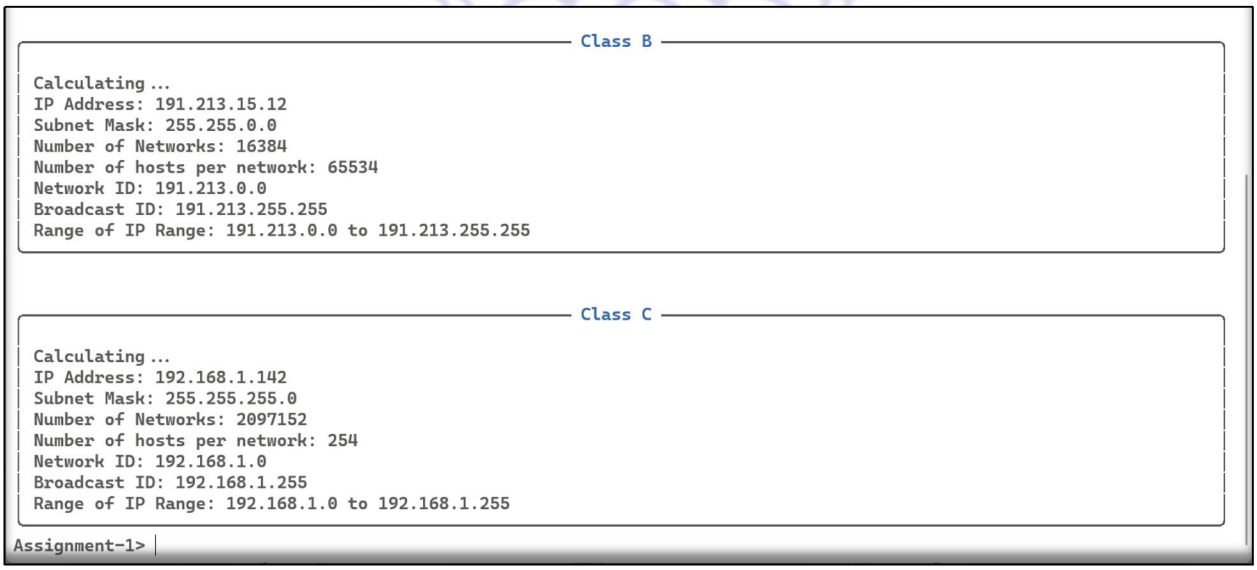
COMPUTER NETWORKS

Sarvesh Anand Mankar
142203013
TY Comp Div-2, T4 Batch


PowerShell
Assignment-1> python .\IP_Classes.py
IP Addresses Description
IPv4 addresses are 32-bit addresses
IPv4 addresses are divided into classes
IPv4 addresses are divided into public and private addresses
IPv4 addresses are divided into network and host addresses
IPv4 addresses are divided into unicast, multicast and broadcast addresses
Class A
Calculating ...
IP Address: 191.213.15.12
Subnet Mask: 255.0.0.0
Number of Networks: 128
Number of hosts per network: 16777214
Network ID: 191.0.0.0
Broadcast ID: 191.255.255.255
Range of IP Range: 191.0.0.0 to 191.255.255.255


Class B
Calculating ...
IP Address: 191.213.15.12
Subnet Mask: 255.255.0.0
Number of Networks: 16384
Number of hosts per network: 65534
Network ID: 191.213.0.0
Broadcast ID: 191.213.255.255
Range of IP Range: 191.213.0.0 to 191.213.255.255
Class C
Calculating ...
IP Address: 192.168.1.142
Subnet Mask: 255.255.255.0
Number of Networks: 2097152
Number of hosts per network: 254
Network ID: 192.168.1.0
Broadcast ID: 192.168.1.255
Range of IP Range: 192.168.1.0 to 192.168.1.255
Assignment-1>

IP Address and Subnet Description

```python
class IPAddresses:
    def __init__(self):
        # Short Single liner descriptions of IPv4
        self.IP_info = [
            "IPv4 addresses are 32-bit addresses",
            "IPv4 addresses are divided into classes",
            "IPv4 addresses are divided into public and private addresses",
            "IPv4 addresses are divided into network and host addresses",
            "IPv4 addresses are divided into unicast, multicast and broadcast addresses",
        ]

        self.class_Info = {
            "A": """Class A:
                - The first bit is always 0
                - The first octet represents the network address
                - The last three octets represent the host address
                - The default subnet mask is 255.0.0.0
            """,
            "B": """Class B:
                - The first two bits are always 10
                - The first two octets represent the network address
                - The last two octets represent the host address
                - The default subnet mask is 255.255.0.0
            """,
            "C": """Class C:
                - The first three bits are always 110
                - The first three octets represent the network address
                - The last octet represents the host address
                - The default subnet mask is 255.255.255.0
            """,
            "D": """Class D:
                - The first four bits are always 1110
                - The address is used for multicasting
            """,
            "E": """Class E:
                - The first four bits are always 1111
                - The address is reserved for experimental purposes
            """,
        }

        self.subnets = {
            "A": "255.0.0.0",
            "B": "255.255.0.0",
            "C": "255.255.255.0",
        }

    def display_ip_info(self):
        string=''
        for info in self.IP_info:
            string+=info+"\n"
        return string

    def display_class_info(self, className: str):
        return self.class_Info[className]
```

Solution Code

```python
class IPClasses(IPAddresses):
    def __init__(self, className: str):
        self.className = className
        super().__init__()

        if self.className not in self.subnets.keys():
            print("Invalid Class Name")
            exit(0)

        if self.className == "A":
            self.bitReserved = 1
        elif self.className == "B":
            self.bitReserved = 2
        elif self.className == "C":
            self.bitReserved = 3
        else:
            self.bitReserved = 4

    def calculate(self, ip_address: str):
        string = "\nCalculating..."
        string += "\nIP Address: " + ip_address
        string += "\nSubnet Mask: " + self.subnets[self.className]

        # Calculating number of Networks
        subnet_mask = self.subnets[self.className].split(".")
        n1 = 0
        for i in subnet_mask:
            if i == "255":
                n1 += 8

        number_of_networks = 2 ** (n1 - self.bitReserved)
        string += "\nNumber of Networks: " + str(number_of_networks)

        # Calculating number of hosts per network
        number_of_hosts = 2 ** (32 - n1) - 2  # -2 for network and broadcast address
        string += "\nNumber of hosts per network: " + str(number_of_hosts)

        # Calculating Network ID and Broadcast ID
        ip_address = ip_address.split(".")
        network_id = []
        broadcast_id = []
        for i in range(4):
            if i < self.bitReserved:
                network_id.append(ip_address[i])
                broadcast_id.append(ip_address[i])
            else:
                network_id.append("0")
                broadcast_id.append("255")

        string += "\n" + "Network ID: " + ".".join(network_id)
        string += "\n" + "Broadcast ID: " + ".".join(broadcast_id)

        # Calculating Range of IP Range
        ip_range = []
        for i in range(4):
            if i < self.bitReserved:
                ip_range.append(ip_address[i])
            else:
                ip_range.append("0")

        string += (
            "\nRange of IP Range: "
            + ".".join(ip_range)
            + " to "
            + ".".join(broadcast_id)
        )
        return string
```