Assignment 8

Name: Hitesh Tolani

Roll no: 73

Class: SY-AIDS-A

Title: Write a program to demonstrate subnetting and find the subnet masks

Theory:

Subnet masks are a fundamental aspect of IP addressing in computer networks, serving as the blueprint for dividing large networks into smaller, more manageable subnetworks, or subnets. At its core, a subnet mask is a 32-bit value that works in tandem with IP addresses to delineate the boundary between the network and host portions of an address. Conceptually, subnet masks consist of a sequence of contiguous 1s followed by a sequence of contiguous 0s, with each 1 indicating a bit reserved for the network portion and each 0 representing a bit available for host addressing. By performing a bitwise AND operation between an IP address and its corresponding subnet mask, network administrators can swiftly ascertain the network ID, revealing the specific subnet to which an IP address belongs. This partitioning of networks into smaller subnets not only enhances network efficiency by minimizing broadcast domains and optimizing traffic flow but also bolsters network security by confining broadcast traffic and segregating different segments of the network.

Code:

```
ip_subnet_map = {
    "A":[0,127,"255.0.0.0"],
    "B":[128,191,"255.255.0.0"],
    "C":[192,223,"255.255.255.0"],
    "D":[224,239,"255.255.0.0"],
    "E":[240,255,"255.255.0.0"],
}

def is_ip_valid(ip_add:str) -> bool:
    subnets = ip_add.split(".")
    for subnet in subnets:
        if not(0 <= int(subnet) <= 255):
            return False

    return True


def get_ip_and_subnet_mask(first_octet:int):
    subnet_mask = None
    ip_class = None
```

```
    for k,v in ip_subnet_map.items():
        if(v[0] <= first_octet <= v[1]):
            subnet_mask = v[2]
            ip_class = k
            return subnet_mask,ip_class

    return None,None


if __name__ == "__main__":
    ip_add = input("Enter IP\n")

    if(is_ip_valid(ip_add)):
        first_octet = int(ip_add[:3])
        subnet_mask,ip_class = get_ip_and_subnet_mask(first_octet)
        print(f"IP Address class is {ip_class} \nSubnet mask is
{subnet_mask}")
    else:
        print("Invalid IP address")
```

Output:

Enter IP

255.0.1.1

IP Address class is E

Subnet mask is 255.255.0.0

Conclusion:

In this assignment, we have learned about subnetting and successfully implemented a program about the same.