**//CODE:**

def prefix\_to\_subnet\_mask(prefix\_length):

mask = (0xFFFFFFFF >> (32 - prefix\_length)) << (32 - prefix\_length)

return [

(mask >> 24) & 0xFF,

(mask >> 16) & 0xFF,

(mask >> 8) & 0xFF,

mask & 0xFF

]

def ip\_class(first\_octet):

first\_octet = int(first\_octet)

if 1 <= first\_octet <= 126:

return "Class A"

elif 128 <= first\_octet <= 191:

return "Class B"

elif 192 <= first\_octet <= 223:

return "Class C"

elif 224 <= first\_octet <= 239:

return "Class D (Multicast)"

elif 240 <= first\_octet <= 255:

return "Class E (Reserved)"

else:

return "Unknown"

def main():

ip = input("Enter IP address (e.g., 192.168.1.0): ")

prefix = int(input("Enter prefix length (e.g., 24 for /24): "))

mask = prefix\_to\_subnet\_mask(prefix)

first\_octet = ip.split('.')[0]

total\_ips = 2 \*\* (32 - prefix)

usable\_hosts = total\_ips - 2 if prefix < 31 else total\_ips # special cases for /31 and /32

print(f"IP Class: {ip\_class(first\_octet)}")

print(f"Subnet Mask: {'.'.join(map(str, mask))}")

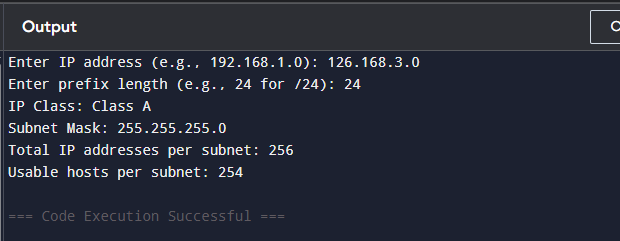
print(f"Total IP addresses per subnet: {total\_ips}")

print(f"Usable hosts per subnet: {usable\_hosts}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**//OUTPUT:**

****