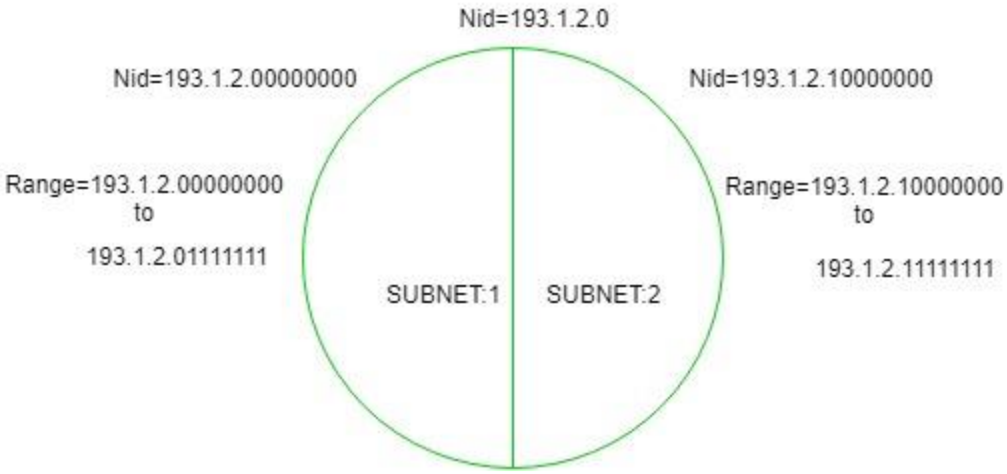
When a bigger network is divided into smaller networks, to maintain security, then that is known as Subnetting. So, maintenance is easier for smaller networks. For example, if we consider a class A address, the possible number of hosts is 2^{24} for each network, it is obvious that it is difficult to maintain such a huge number of hosts, but it would be quite easier to maintain if we divide the network into small parts.

Uses of Subnetting

1. Subnetting helps in organizing the network in an efficient way which helps in expanding the technology for large firms and companies.
2. Subnetting is used for specific staffing structures to reduce traffic and maintain order and efficiency.
3. Subnetting divides domains of the broadcast so that traffic is routed efficiently, which helps in improving network performance.
4. Subnetting is used in increasing network security.

The network can be divided into two parts: To divide a network into two parts, you need to choose one bit for each Subnet from the host ID part.



In class C the first 3 octets are network bits so it remains as it is.

- **For Subnet-1:** The first bit which is chosen from the host id part is zero and the range will be from (193.1.2.00000000 till you get all 1's in the host ID part i.e, 193.1.2.01111111) except for the first bit which is chosen zero for subnet id part.

Thus, the range of subnet 1 is: **193.1.2.0 to 193.1.2.127**

```
Subnet id of Subnet-1 is : 193.1.2.0
The direct Broadcast id of Subnet-1 is: 193.1.2.127
The total number of hosts possible is: 126 (Out of 128,
2 id's are used for Subnet id & Direct Broadcast id)
The subnet mask of Subnet- 1 is: 255.255.255.128
```

- **For Subnet-2:** The first bit chosen from the host id part is one and the range will be from (193.1.2.100000000 till you get all 1's in the host ID part i.e, 193.1.2.11111111).

Thus, the range of subnet-2 is: **193.1.2.128 to 193.1.2.255**

```
Subnet id of Subnet-2 is : 193.1.2.128
The direct Broadcast id of Subnet-2 is: 193.1.2.255
The total number of hosts possible is: 126 (Out of 128,
2 id's are used for Subnet id &  Direct Broadcast id)
The subnet mask of Subnet- 2 is: 255.255.255.128
The best way to find out the subnet mask of a subnet
is to set the fixed bit of host-id to 1 and the rest to 0.
```

Finally, after using the subnetting the total number of usable hosts is reduced from 254 to 252.

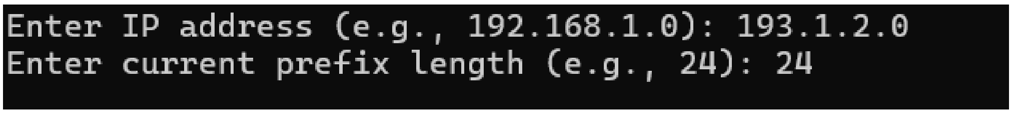
Note:

1. To divide a network into four (2_2) parts you need to choose two bits from the host id part for each subnet i.e, (00, 01, 10, 11).
2. To divide a network into eight (2_3) parts you need to choose three bits from the host id part for each subnet i.e, (000, 001, 010, 011, 100, 101, 110, 111) and so on.
3. We can say that if the total number of subnets in a network increases the total number of usable hosts decreases.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Function to convert an IP address to a 32-bit integer
unsigned int ipToInt(char* ip) {
   unsigned int a, b, c, d;
   sscanf(ip, "%u.%u.%u.%u", &a, &b, &c, &d);
   return (a << 24) | (b << 16) | (c << 8) | d;
}

// Function to convert a 32-bit integer to an IP address
void intToIp(unsigned int ip, char* buffer) {
   sprintf(buffer, "%u.%u.%u.%u", (ip >> 24) & 0xFF, (ip >> 16) & 0xFF, (ip >> 8) & 0xFF, ip & 0xFF);
}

// Function to calculate the subnet mask from a prefix length
unsigned int calculateSubnetMask(int prefixLength) {
   return prefixLength == 0 ? 0 : ~((1 << (32 - prefixLength)) - 1);
}

int main() {
   char ip[16];
```

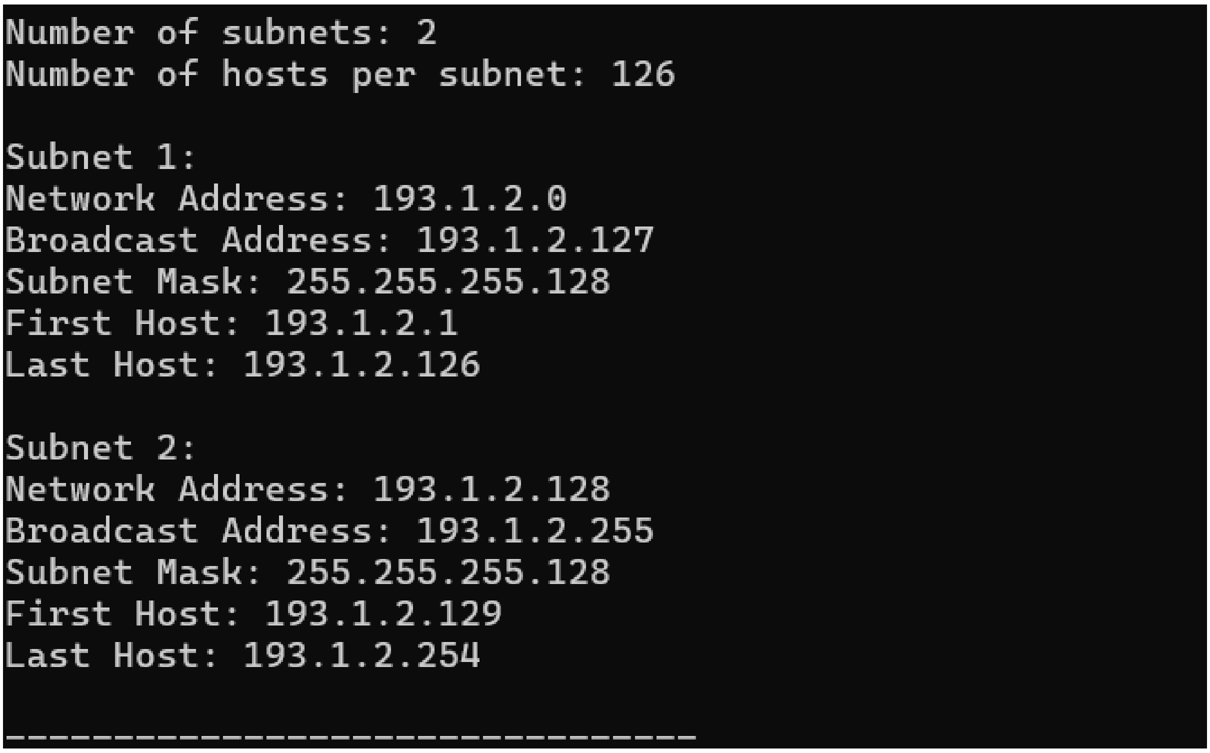
```
    int prefixLength, newPrefixLength;
    unsigned int subnetMask, newSubnetMask, ipInt;
    char buffer[16];

    // Input IP address and prefix length
    printf("Enter IP address (e.g., 192.168.1.0): ");
    scanf("%s", ip);
    printf("Enter current prefix length (e.g., 24): ");
    scanf("%d", &prefixLength);

    // New prefix length for creating two subnets
    newPrefixLength = prefixLength + 1;

    // Convert IP address to integer
    ipInt = ipToInt(ip);

    // Calculate original subnet mask and new subnet mask
    subnetMask = calculateSubnetMask(prefixLength);
    newSubnetMask = calculateSubnetMask(newPrefixLength);

    // Calculate the number of hosts per subnet
    int hostsPerSubnet = (1 << (32 - newPrefixLength)) - 2; // subtract 2 for
network and broadcast addresses

    printf("\nNumber of subnets: 2\n");
    printf("Number of hosts per subnet: %d\n", hostsPerSubnet);

    // Generate subnets
    for (int i = 0; i < 2; i++) {
        unsigned int subnetNetwork = (ipInt & subnetMask) | (i << (32 -
newPrefixLength));
        unsigned int subnetBroadcast = subnetNetwork | ~newSubnetMask;
        unsigned int firstHost = subnetNetwork + 1;
        unsigned int lastHost = subnetBroadcast - 1;

        printf("\nSubnet %d:\n", i + 1);
        printf("Network Address: ");
        intToIp(subnetNetwork, buffer);
        printf("%s\n", buffer);

        printf("Broadcast Address: ");
        intToIp(subnetBroadcast, buffer);
        printf("%s\n", buffer);

        printf("Subnet Mask: ");
        intToIp(newSubnetMask, buffer);
        printf("%s\n", buffer);

        printf("First Host: ");
        intToIp(firstHost, buffer);
        printf("%s\n", buffer);

        printf("Last Host: ");
        intToIp(lastHost, buffer);
        printf("%s\n", buffer);
    }

    return 0;
}
```

INPUT:



```
Enter IP address (e.g., 192.168.1.0): 193.1.2.0
Enter current prefix length (e.g., 24): 24
```

OUTPUT:



```
Number of subnets: 2
Number of hosts per subnet: 126

Subnet 1:
Network Address: 193.1.2.0
Broadcast Address: 193.1.2.127
Subnet Mask: 255.255.255.128
First Host: 193.1.2.1
Last Host: 193.1.2.126

Subnet 2:
Network Address: 193.1.2.128
Broadcast Address: 193.1.2.255
Subnet Mask: 255.255.255.128
First Host: 193.1.2.129
Last Host: 193.1.2.254

--------------------------------
```