Lab Experiment 7

Write a program to display the class of IP address, network mask and generate the subnet IP address based on the subnet bits entered from the keyboard.

Program:

```
#include <iostream>
#include <string>
#include <vector>
#include <cmath>

using namespace std;

char getClass(const string &ip)
{
    int firstOctet = stoi(ip.substr(0, ip.find('.')));
    if (firstOctet >= 1 && firstOctet <= 126)
    {
        return 'A';
    }
    else if (firstOctet >= 128 && firstOctet <= 191)
    {
        return 'B';
    }
    else if (firstOctet >= 192 && firstOctet <= 223)
    {
        return 'C';
    }
    else if (firstOctet >= 224 && firstOctet <= 239)
    {
        return 'D';
    }
    else if (firstOctet >= 240 && firstOctet <= 255)
    {
        return 'E';
    }
    else
    {
        return 'Z';
    }
}

pair<string, string> getSubnetMaskandSubnetIP(string ipAddress, int
subnetBits, char ipClass)
{
    pair<string, string> result;
    vector<int> octets;
    size_t i = 0;
```

```
    while ((i = ipAddress.find('.')) != string::npos)
    {
        octets.push_back(stoi(ipAddress.substr(0, i)));
        ipAddress.erase(0, i + 1);
    }
    octets.push_back(stoi(ipAddress));
    int subnetMask;
    switch (ipClass)
    {
    case 'A':
        subnetMask = (0xFFFFFFFF << (32 - (8 + subnetBits)) & 0xFFFFFFFF);
        break;
    case 'B':
        subnetMask = (0xFFFFFFFF << (32 - (16 + subnetBits)) & 0xFFFFFFFF);
        break;
    case 'C':
        subnetMask = (0xFFFFFFFF << (32 - (24 + subnetBits)) & 0xFFFFFFFF);
        break;
    case 'D':
        subnetMask = (0xFFFFFFFF << (32 - (subnetBits)) & 0xFFFFFFFF);
        break;
    case 'E':
        subnetMask = (0xFFFFFFFF << (32 - (subnetBits)) & 0xFFFFFFFF);
        break;
    default:
        cout << "Improper class for subnetting." << endl;
        break;
    }
    string subnetMaskStr = to_string(subnetMask >> 24 & 0xFF) + "." +
                           to_string((subnetMask >> 16) & 0xFF) + "." +
                           to_string((subnetMask >> 8) & 0xFF) + "." +
                           to_string(subnetMask & 0xFF);
    result.first = subnetMaskStr;

    string subnetIPStr = to_string(subnetMask >> 24 & octets[0]) + "." +
                         to_string((subnetMask >> 16) & 0xFF & octets[1]) +
"." +
                         to_string((subnetMask >> 8) & 0xFF & octets[2]) + "."
+
                         to_string(subnetMask & 0xFF & octets[3]);
    result.second = subnetIPStr;
    return result;
}

int main()
{
    string ipAddress;
    int subnetBits;
```

```
    cout << "Enter IP Address (Format: xxx.xxx.xxx.xxx): ";
    cin >> ipAddress;

    char ipClass = getClass(ipAddress);
    cout << "IP Address Class: " << ipClass << endl;

    cout << "Enter Subnet Bits: ";
    cin >> subnetBits;

    pair<string, string> result = getSubnetMaskandSubnetIP(ipAddress,
subnetBits, ipClass);
    cout << "Subnet Mask: ";
    cout << result.first << endl;
    cout << "Subnet IP Address: ";
    cout << result.second << endl;
    return 0;
}
```

Output:

```
Enter IP Address (Format: xxx.xxx.xxx.xxx): 129.134.38.37
IP Address Class: B
Enter Subnet Bits: 5
Subnet Mask: 255.255.248.0
Subnet IP Address: 129.134.32.0
PS D:\SRM AP\SEM 5\Computer Networks Lab\Week7>
```