

Code:

```
#include <bits/stdc++.h>

using namespace std;

vector<string> split(const string &s, char delimiter)
{
    vector<string> tokens;
    string token;
    istringstream tokenStream(s);
    while (getline(tokenStream, token, delimiter))
    {
        tokens.push_back(token);
    }
    return tokens;
}

int main()
{
    string cidr;
    string ip_address;

    cout << "Enter IP Address: " << endl;
    cin >> ip_address;

    cout << "Enter the subnet mask in CIDR notation (e.g., /24):" << endl;
    cin >> cidr;

    // Split the IP address into octets
    vector<string> ip_octets = split(ip_address, '.');

    // Extract the CIDR prefix length
```

```
int cidr_prefix_length = stoi(cidr.substr(1));

// Calculate the subnet mask
uint32_t subnet_mask_value = 0xFFFFFFFFU << (32 - cidr_prefix_length);

// Convert octets to integers
uint32_t ip_address_value = (stoi(ip_octets[0]) << 24) |
    (stoi(ip_octets[1]) << 16) |
    (stoi(ip_octets[2]) << 8) |
    stoi(ip_octets[3]);

// Calculate the network address
uint32_t network_address = ip_address_value & subnet_mask_value;

// Calculate the broadcast address
uint32_t broadcast_address = network_address | (~subnet_mask_value);

int network_octets[4];
int broadcast_octets[4];

for (int i = 0; i < 4; ++i)
{
    network_octets[i] = (network_address >> (24 - 8 * i)) & 0xFF;
    broadcast_octets[i] = (broadcast_address >> (24 - 8 * i)) & 0xFF;
}

// Output the initial and end addresses
cout << "Initial Address: " << network_octets[0] << "." << network_octets[1] << "." << network_octets[2] << "." <<
network_octets[3] << endl;

cout << "End Address: " << broadcast_octets[0] << "." << broadcast_octets[1] << "." << broadcast_octets[2] << "." <<
broadcast_octets[3] << endl;
```

```
// cout << "Initial Address: " << network_address<<endl;
// cout << "End Address: " << broadcast_address<<endl;

return 0;
}
```

Output:

Enter IP Address:

201.1.1.0

Enter the subnet mask in CIDR notation (e.g., /24):

/27

Initial Address: 201.1.1.0

End Address: 201.1.1.31