**LAB ASSIGNMENT – 4**

Grihit Budhiraja

19BCE2141

**Q1. First Address and Last Address of an IP Address**

**Code –**

#include <bits/stdc++.h>

using namespace std;

int binaryToDecimal(int n[])

{

int i,num=0,base=1;

for(i=7;i>=0;i--)

{

num = num + base\*n[i];

base = base \* 2;

}

return num;

}

int main()

{

int ip[4],n,i,num1,num2,mask;

cout<<"\nEnter the ip address seperated by a space: ";

for(i=0;i<4;i++)

cin>>ip[i];

cout<<"\nEnter the mask: ";

cin>>mask;

int temp1=ip[3];

int bNum1[32];

int start=0,end=7;

i=0;

for(int k=0;k<8;k++)

{

if(temp1>0)

{

bNum1[i] = temp1 % 2;

temp1 = temp1 / 2;

i++;

}

else

{

bNum1[i]=0;

i++;

}

}

while (start < end)

{

int temp = bNum1[start];

bNum1[start] = bNum1[end];

bNum1[end] = temp;

start++;

end--;

}

int m = 32 - mask;

int temp2[32],temp3[32];

for(i=0;i<7;i++)

{

temp2[i]=bNum1[i];

temp3[i]=bNum1[i];

}

for(i=7;i>=m;i--)

{

temp2[i]=0;

}

for(i=7;i>=m;i--)

{

temp3[i]=1;

}

num1=binaryToDecimal(temp2);

num2=binaryToDecimal(temp3);

cout<<"\nFirst Address is ";

for(i=0;i<3;i++)

cout<<ip[i]<<".";

cout<<num1;

cout<<"\nLast Address is ";

for(i=0;i<3;i++)

cout<<ip[i]<<".";

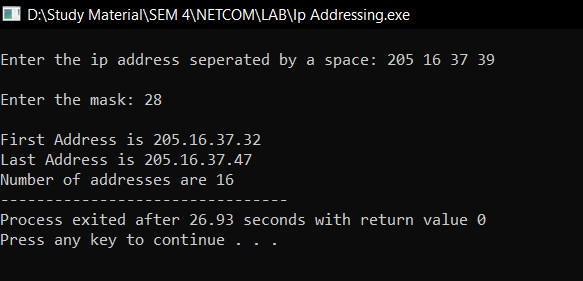
cout<<num2;

cout<<"\nNumber of addresses are "<<pow(2,m);

return 0;

}

**Output –**



**Q2. With AND operation**

**Code –**

#include <bits/stdc++.h>

using namespace ::std;

string binDec = "";

string decBin = "";

string maskString = "";

int convertMask(int n)

{

int num = n;

int dec\_value = 0;

int base = 1;

int temp = num;

while (temp)

{

int last\_digit = temp % 10;

temp = temp / 10;

dec\_value += last\_digit \* base;

base = base \* 2;

}

return dec\_value;

}

string complement(string bits)

{

for (int i = 0; i < bits.length(); i++)

{

if (bits[i] == '1')

{

bits[i] = '0';

}

else

{

bits[i] = '1';

}

}

return bits;

}

void binToDecimal(string n)

{

string num = n;

int dec\_value = 0;

int base = 1;

int len = num.length();

for (int i = len - 1; i >= 0; i--)

{

if (num[i] == '1')

dec\_value += base;

base = base \* 2;

}

stringstream ss;

ss << dec\_value;

string s;

ss >> s;

binDec = binDec + s + '.';

}

void FirstAddress(string bits, string mask)

{

for (int i = 0; i < 32; i++)

{

if (bits[i] != mask[i])

{

bits[i] = '0';

}

}

int j = 0;

string block = "";

for (int i = 0; i <= 31; i++)

{

block += bits[i];

if ((i + 1) % 8 == 0)

{

binToDecimal(block);

block = "";

}

}

binDec[binDec.length() - 1] = ' ';

cout << "First Address is : " << binDec << endl;

binDec = "";

}

void LastAddress(string bits, string mask)

{

mask = complement(mask);

for (int i = 0; i < 32; i++)

{

if (bits[i] == '0' && mask[i] == '1')

{

bits[i] = '1';

}

}

int j = 0;

string block = "";

for (int i = 0; i <= 31; i++)

{

block += bits[i];

if ((i + 1) % 8 == 0)

{

binToDecimal(block);

block = "";

}

}

binDec[binDec.length() - 1] = ' ';

cout << "Last Address is : " << binDec << endl;

binDec = "";

}

void NoOfAddress(string bits, string mask)

{

mask = complement(mask);

int maskNum = atoi(mask.c\_str());

maskNum = convertMask(maskNum);

cout << "Number of addresses are : " << maskNum + 1 << endl;

}

int decToBinary(int n)

{

for (int i = 7; i >= 0; i--)

{

int k = n >> i;

if (k & 1)

decBin += "1";

else

decBin += "0";

}

}

int main()

{

string IP;

cout << "Enter IP address ";

string block = "";

cin >> IP;

string maskBits = "";

maskBits = maskBits + IP[IP.length() - 2] + IP[IP.length() - 1];

int mask = atoi(maskBits.c\_str());

for (int x = 0; x < mask; x++)

{

maskString += '1';

}

int remBits = 32 - mask;

while (remBits)

{

maskString += '0';

remBits--;

}

int i = 0, ctr = 0, j = 0;

while (i < IP.length() - 3)

{

if (IP[i] != '.')

{

block += IP[i];

}

if (IP[i] == '.' || IP[i] == IP[IP.length() - 4])

{

int b = atoi(block.c\_str());

decToBinary(b);

block = " ";

}

i++;

}

FirstAddress(decBin, maskString);

LastAddress(decBin, maskString);

NoOfAddress(decBin, maskString);

}

**Output –**

