Задача 11

Разработать программу, вычисляющую по IP-адресу и маске сети NETWORK ID и HOST ID.

Вани(С#)

using System;

namespace \_11.IP\_M\_NetWork\_host

{

class Program

{

static void Main(string[] args)

{

//Разработать программу, вычисляющую по IP-адресу и маске сети NETWORK ID и HOST ID.

(string IP, string Mask) la = ("192.2.172.1", "255.255.128.0");

string[] IPcon = la.IP.Split('.');

int[] IPoct = new int[4];

foreach (string i in IPcon)

{

Console.Write(i + ".");

}

for (int i = 0; i < IPcon.Length; i++)

{

IPoct[i] = Convert.ToInt32(IPcon[i]);

}

Console.WriteLine();

string[] Maskcon = la.Mask.Split('.');

int[] Maskoct = new int[4];

foreach (string i in Maskcon)

{

Console.Write(i + ".");

}

for (int i = 0; i < Maskcon.Length; i++)

{

Maskoct[i] = Convert.ToInt32(Maskcon[i]);

}

int[] NetWorkID = new int[4];

for (int i = 0; i < Maskoct.Length; i++)

{

NetWorkID[i] = IPoct[i] & Maskoct[i];

}

int[] HostID = new int[4];

for (int i = 0; i < Maskoct.Length; i++)

{

HostID[i] = IPoct[i] & ~Maskoct[i];

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("HostID:");

foreach (int i in HostID)

{

Console.Write(i + ".");

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("NetWorkID:");

foreach (int i in NetWorkID)

{

Console.Write(i + ".");

}

Console.ReadKey();

}

}

}

ИЗ ТОГО ФАЙЛА(С++)

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

bool check(int x, int x1, int x2, int x3) {

int arr[8], arr2[8], arr3[8], arr4[8];

int flag = 0;

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

{

arr[i] = ((int)((x >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i < 7) {

if ((arr[i] == 1) && (arr[i + 1] == 0)) {

cout << "Маска неверная" << endl;

break;

flag = flag + 1;

}

}

}

//

if (flag > 0) {

return 0;

}

else {

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

{

arr2[i] = ((int)((x1 >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i == 0) {

if (arr[0] == 0 && arr2[0]) {

cout << "Маска неверная" << endl;

break;

}

else {

if (i < 7) {

if ((arr2[i] == 1) && (arr2[i + 1] == 0)) {

cout << "Маска неверная" << endl;

break;

flag = flag + 1;

}

}

}

}

}

//3 октет

if (flag > 0) {

return 0;

}

else {

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

{

arr3[i] = ((int)((x2 >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i == 0) {

if (arr2[0] == 0 && arr3[0]) {

cout << "Маска неверная" << endl;

break;

}

else {

if (i < 7) {

if ((arr3[i] == 1) && (arr3[i + 1] == 0)) {

cout << "Маска неверная" << endl;

break;

flag = flag + 1;

}

}

}

}

}

//4 октет

if (flag > 0) {

return 0;

}

else {

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

{

arr4[i] = ((int)((x3 >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i == 0) {

if (arr3[0] == 0 && arr4[0]) {

cout << "Маска неверная" << endl;

break;

}

else {

if (i < 7) {

if ((arr4[i] == 1) && (arr4[i + 1] == 0)) {

cout << "Маска неверная" << endl;

break;

flag = flag + 1;

}

}

}

}

}

}

}

if (flag > 0) {

return 0;

}

else {

return 1;

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

string str;

cout << "Введите маску" << endl;

cin >> str;

string str\_x0, str\_x1, str\_x2, str\_x3;

int flag = 0;

for (int i = 0; i < str.size(); i++) {

if (str[i] == '.') {

flag = flag + 1;

}

if (flag == 0) {

if (str[i] != '.') {

str\_x0 = str\_x0 + str[i];

}

}

if (flag == 1) {

if (str[i] != '.') {

str\_x1 = str\_x1 + str[i];

}

}

if (flag == 2) {

if (str[i] != '.') {

str\_x2 = str\_x2 + str[i];

}

}

if (flag == 3) {

if (str[i] != '.') {

str\_x3 = str\_x3 + str[i];

}

}

}

//переведем в целочисленный

int x0 = atoi(str\_x0.c\_str()), x1 = atoi(str\_x1.c\_str()), x2 = atoi(str\_x2.c\_str()), x3 = atoi(str\_x3.c\_str());

int ch = check(x0, x1, x2, x3);

if (ch == 1) {

//IP

string str1;

cout << "Введите IP" << endl;

cin >> str1;

string str\_y0, str\_y1, str\_y2, str\_y3;

flag = 0;

for (int i = 0; i < str1.size(); i++) {

if (str1[i] == '.') {

flag = flag + 1;

}

if (flag == 0) {

if (str1[i] != '.') {

str\_y0 = str\_y0 + str1[i];

}

}

if (flag == 1) {

if (str1[i] != '.') {

str\_y1 = str\_y1 + str1[i];

}

}

if (flag == 2) {

if (str1[i] != '.') {

str\_y2 = str\_y2 + str1[i];

}

}

if (flag == 3) {

if (str1[i] != '.') {

str\_y3 = str\_y3 + str1[i];

}

}

}

//переведем в целочисленный

int y0 = atoi(str\_y0.c\_str()), y1 = atoi(str\_y1.c\_str()), y2 = atoi(str\_y2.c\_str()), y3 = atoi(str\_y3.c\_str());

cout << "NETWORK ID" << endl;

cout << (y0 & x0) << ".";

cout << (y1 & x1) << ".";

cout << (y2 & x2) << ".";

cout << (y3 & x3) << endl;

cout << "HOST ID" << endl;

x0 = ~x0; x1 = ~x1; x2 = ~x2; x3 = ~x3;

cout << (y0 | x0) \* (-1) << ".";

cout << (y1 | x1) \* (-1) << ".";

cout << (y2 | x2) \* (-1) << ".";

cout << (y3 | x3) \* (-1) << endl;

}

else {

cout << "Маска неверная, перезапустите программу" << endl;

}

}

Задача 12

Разработать программу, проверяющую маску на непрерывность единиц (в двоичной форме). Ввод маски осуществлять в диалоговом режиме.

ВАНЯ(С#)

using System;

namespace \_12.проверка\_на\_непрерывность

{

class Program

{

static void Main(string[] args)

{

string mask = "255.255.128.0";

Console.WriteLine("Введите маску в формате " + mask);

mask = Console.ReadLine();

string[] maskcon = mask.Split('.');

int[] maskoct = new int[4];

bool Pre = true;

foreach (string i in maskcon)

{

Console.Write(i + ".");

}

for (int i = 0; i < maskcon.Length; i++)

{

maskoct[i] = Convert.ToInt32(maskcon[i]);

}

bool k = true;

for (int i = 0; i < maskoct.Length; i++)

{

if (k == false && maskoct[i] != 0)

Pre = false;

if (maskoct[i] == 255) continue;

if (maskoct[i] == 254 || maskoct[i] == 252 || maskoct[i] == 248 || maskoct[i] == 240 || maskoct[i] == 224 || maskoct[i] == 192 || maskoct[i] == 128 || maskoct[i] == 0)

{

k = false;

continue;

}

Pre = false;

}

Console.WriteLine();

if (Pre)

Console.WriteLine("маска непрерывна");

else

Console.WriteLine("ОШИБКА. Маска прерывна");

Console.ReadKey();

}

}

}

Задача 13

Разработать программу, проверяющую IP-адреса на разрешенные символы (IP-адрес состоит только из цифр – все остальные символы запрещены; октеты должны быть разделены точками – другие варианты недопустимы). Количество октетов не проверять. Ввод IP-адреса осуществлять в диалоговом режиме.

ИЗ ТОГО ФАЙЛА(С++)

#include <iostream>

#include <string>

#include <vector>

#include <sstream>

using namespace std;

int main()

{

string n,k;

cout << "Vvedite ip-adress: ";

cin >> n;

cout << "ip:" << n << endl;

// проверяет корректность символов при вводе Ip

for (int i = 0; i<= n.length()-1; i++)

{

if (n[i] == '0' || n[i] == '1' || n[i] == '2' || n[i] == '3' ||

n[i] == '4' || n[i] == '5' || n[i] == '6' || n[i] == '7' ||

n[i] == '8' || n[i] == '9' || n[i] == '.' )

{

k ="good";

}

else

{

k = "bad";

break;

}

}

cout << k<<endl;

}

Задача 14

Разработать программу, проверяющую IP-адреса на количество октетов и их допустимые значения. Также обязательно проверить символы, разделяющие октеты и их количество. Ввод IP-адреса осуществлять в диалоговом режиме.

ИЗ ТОГО ФАЙЛА(С++)

#include <iostream>

#include <string>

#include <vector>

#include <sstream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

string n, k;

cout <<"Введите ip: ";

cin>> n;

cout<< "ip:" << n << endl;

// проверяет корректность символов при вводе Ip

for (int i = 0; i <= n.length() - 1; i++)

{

if (n[i] == '0' || n[i] == '1' || n[i] == '2' || n[i] == '3' ||

n[i] == '4' || n[i] == '5' || n[i] == '6' || n[i] == '7' ||

n[i] == '8' || n[i] == '9' || n[i] == '.')

{

k = "ip-адрес содержит только разрешённые символы";

}

else

{

k = " В ip-адресе есть запрещённые символы";

break;

}

}

cout << k << endl;

}

Задача 15

Разработать программу, рассчитывающую возможное количество адресов (подсетей) в сети по заданному IP-адресу и маске (ввести в диалоговом режиме).

ВАНЯ(С#)

using System;

namespace \_15.Возможное\_колич\_адресов\_в\_сети\_по\_заданному\_ip\_m

{

class Program

{

static void Main(string[] args)

{

//Разработать программу, рассчитывающую возможное количество адресов в сети по заданному IP-адресу и маске (ввести в диалоговом режиме).

(string IP, string Mask) la = ("192.2.172.1", "255.255.128.0");

Console.WriteLine("Введите IP в формате " + la.IP);

la.IP = Console.ReadLine();

Console.WriteLine("Введите маску в формате " + la.Mask);

la.Mask = Console.ReadLine();

string[] Maskcon = la.Mask.Split('.');

int[] Maskoct = new int[4];

int stepen = 0;

for (int i = 0; i < Maskcon.Length; i++)

{

Maskoct[i] = Convert.ToInt32(Maskcon[i]);

}

foreach (int i in Maskoct)

{

Console.WriteLine(i);

if (i == 255) stepen += 0;

if (i == 254) stepen += 1;

if (i == 252) stepen += 2;

if (i == 248) stepen += 3;

if (i == 240) stepen += 4;

if (i == 224) stepen += 5;

if (i == 192) stepen += 6;

if (i == 128) stepen += 7;

if (i == 0) stepen += 8;

}

double yzl = 0;

yzl = Math.Pow(Convert.ToDouble(2), Convert.ToDouble(stepen)) - 2;

if (yzl > 1)

Console.WriteLine("возможное количество адресов в сети " + yzl);

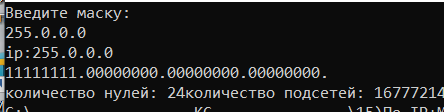
Console.ReadKey();

}

}

}

Санин код(С++)



#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

//Разработать программу, рассчитывающую возможное количество адресов(подсетей)

//в сети по заданному IP - адресу и маске(ввести в диалоговом режиме).

//перевод из 10 в 2

int main()

{

setlocale(LC\_ALL, "Russian");

string n, k;

int f = 0;

int d = 0;

int mass[4];

//255.255.240.0

cout << "Введите маску: " << endl;

cin >> n;

cout << "ip:" << n << endl;

//устанавливаем в конце точку чтобы разделить на октеты

if (n[n.length() - 1] != '.')

{

n = n + '.';

}

for (int l = 0; l <= n.length(); l++)

{

if (n[l] == '.')

{

//разделение октетов по точке

k = n.substr(l - f, f);

f = 0;

// переносим string в int и проверяем на число, если число больше 255 то ошибка. Так как в ip(маска) максимальное число 255

int x = atoi(k.c\_str());

if (x > 255)

{

cout << "oktet bolshe 255: " << x << endl;

break;

}

//выводим все октеты

// cout << k << endl;

mass[d] = x;

d += 1;

}

//провека и подсчет чисел до точки

if (n[l] == '0' || n[l] == '1' || n[l] == '2' || n[l] == '3' || n[l] == '4' || n[l] == '5' || n[l] == '6' || n[l] == '7' || n[l] == '8' || n[l] == '9')

{

f += 1;

}

}

//вызов перевода в 2, используем массив и занесение данных в двумерный массив

int i1,i2;

int mass\_2[4][8];

for (i2 = 0; i2<=3;i2++)

{

for (i1 = 7; i1 >= 0; i1--)

{

if ((1 << i1) & mass[i2])

{

cout << "1";

mass\_2[i2][i1] = 1;

}

else

{

cout << "0";

mass\_2[i2][i1] = 0;

}

}

cout << ".";

}

cout << endl;

int y=0;

//вызываем массив(он уже поменял цифры месами в прошлом цикле) и считаем нули до первой 1

for (int i = 3; i >= 0; i--)

{

for (int x =0; x<=7;x++)

{

if (mass\_2[i][x] == 0)

{

y += 1;

}

else

{

break;

}

}

}

cout<<"количество нулей: " << y;

int v = pow(2, y) - 2;

cout << "количество подсетей: " << v;

}

Задача 16

Разработать программу, определяющую MAC-адреса всех доступных компьютеров сети класса A (IP-адреса в сети выбрать самостоятельно).



#include <iostream>

using namespace std;

#include <winsock2.h>

#include <iphlpapi.h>

#include <stdio.h>

#include <conio.h>

#include <Ws2tcpip.h>

#include <vector>

#include <bitset>

#include <string>

#pragma comment(lib , "iphlpapi.lib")

#pragma comment(lib , "ws2\_32.lib")

void GetMacAddress(unsigned char\*, struct in\_addr);

bool CheckAddress(string ip) {

int points = 0, numbers = 0;

string buff;

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

if (ip[i] <= '9' && ip[i] >= '0') {

if (numbers > 3) return false;

buff.push\_back(ip[i]);

numbers++;

}

else {

if (ip[i] == '.') {

if (stoi(buff) > 255) return false;

if (numbers == 0) return false;

points++;

buff.clear();

numbers = 0;

if (ip[i + 1] == '.') return false;

}

else return false;

}

if (numbers > 3) return false;

if (points > 3) return false;

}

return true;

}

int main()

{

setlocale(LC\_ALL, "RUS");

unsigned char mac[6];

struct in\_addr srcip = { 0 };

struct sockaddr\_in sa;

char ip\_address[32];

bool flag = true;

WSADATA firstsock;

if (WSAStartup(MAKEWORD(2, 2), &firstsock) != 0)

{

cout << "Ошибка инициализации winsock";

cout << WSAGetLastError();

return -1;

}

do

{

if (!flag) cout << "Wrong address" << endl;

cout << "Enter IP: ";

cin >> ip\_address;

} while (!(flag = CheckAddress(ip\_address)));

//преобразование IP адреса другим способом

//srcip.s\_addr = inet\_addr(ip\_address);

inet\_pton(AF\_INET, ip\_address, &(sa.sin\_addr));

//Получение MAC по IP

GetMacAddress(mac, sa.sin\_addr);

//GetMacAddress(mac, srcip);

printf("MAC адрес : %.2X-%.2X-%.2X-%.2X-%.2X-%.2X",

mac[0], mac[1], mac[2], mac[3], mac[4], mac[5]);

printf("\n");

//\_getch();

return 0;

}

void GetMacAddress(unsigned char\* mac, struct in\_addr

destip)

{

DWORD ret;

IPAddr srcip;

ULONG MacAddr[2];

ULONG PhyAddrLen = 6;

int i;

srcip = 0;

//Послать ARP пакет

ret = SendARP((IPAddr)destip.S\_un.S\_addr,

srcip, MacAddr, &PhyAddrLen);

//Преобразовать адрес

if (PhyAddrLen)

{

BYTE\* bMacAddr = (BYTE\*)&MacAddr;

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

{

mac[i] = (char)bMacAddr[i];

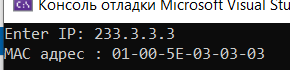
}

}

}

Задача 17

Разработать программу, определяющую MAC-адреса всех доступных компьютеров сети класса B (IP-адреса в сети выбрать самостоятельно).



#include <iostream>

using namespace std;

#include <winsock2.h>

#include <iphlpapi.h>

#include <stdio.h>

#include <conio.h>

#include <Ws2tcpip.h>

#include <vector>

#include <bitset>

#include <string>

#pragma comment(lib , "iphlpapi.lib")

#pragma comment(lib , "ws2\_32.lib")

void GetMacAddress(unsigned char\*, struct in\_addr);

bool CheckAddress(string ip) {

int points = 0, numbers = 0;

string buff;

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

if (ip[i] <= '9' && ip[i] >= '0') {

if (numbers > 3) return false;

buff.push\_back(ip[i]);

numbers++;

}

else {

if (ip[i] == '.') {

if (stoi(buff) > 255) return false;

if (numbers == 0) return false;

points++;

buff.clear();

numbers = 0;

if (ip[i + 1] == '.') return false;

}

else return false;

}

if (numbers > 3) return false;

if (points > 3) return false;

}

return true;

}

int main()

{

setlocale(LC\_ALL, "RUS");

unsigned char mac[6];

struct in\_addr srcip = { 0 };

struct sockaddr\_in sa;

char ip\_address[32];

bool flag = true;

WSADATA firstsock;

if (WSAStartup(MAKEWORD(2, 2), &firstsock) != 0)

{

cout << "Ошибка инициализации winsock";

cout << WSAGetLastError();

return -1;

}

do

{

if (!flag) cout << "Wrong address" << endl;

cout << "Enter IP: ";

cin >> ip\_address;

} while (!(flag = CheckAddress(ip\_address)));

//преобразование IP адреса другим способом

//srcip.s\_addr = inet\_addr(ip\_address);

inet\_pton(AF\_INET, ip\_address, &(sa.sin\_addr));

//Получение MAC по IP

GetMacAddress(mac, sa.sin\_addr);

//GetMacAddress(mac, srcip);

printf("MAC адрес : %.2X-%.2X-%.2X-%.2X-%.2X-%.2X",

mac[0], mac[1], mac[2], mac[3], mac[4], mac[5]);

printf("\n");

//\_getch();

return 0;

}

void GetMacAddress(unsigned char\* mac, struct in\_addr

destip)

{

DWORD ret;

IPAddr srcip;

ULONG MacAddr[2];

ULONG PhyAddrLen = 6;

int i;

srcip = 0;

//Послать ARP пакет

ret = SendARP((IPAddr)destip.S\_un.S\_addr,

srcip, MacAddr, &PhyAddrLen);

//Преобразовать адрес

if (PhyAddrLen)

{

BYTE\* bMacAddr = (BYTE\*)&MacAddr;

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

{

mac[i] = (char)bMacAddr[i];

}

}

}

Задача 18

Разработать программу, определяющую MAC-адреса всех доступных компьютеров сети класса C (IP-адреса в сети выбрать самостоятельно).

#include <iostream>

using namespace std;

#include <winsock2.h>

#include <iphlpapi.h>

#include <stdio.h>

#include <conio.h>

#include <Ws2tcpip.h>

#include <vector>

#include <bitset>

#include <string>

#pragma comment(lib , "iphlpapi.lib")

#pragma comment(lib , "ws2\_32.lib")

void GetMacAddress(unsigned char\*, struct in\_addr);

bool CheckAddress(string ip) {

int points = 0, numbers = 0;

string buff;

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

if (ip[i] <= '9' && ip[i] >= '0') {

if (numbers > 3) return false;

buff.push\_back(ip[i]);

numbers++;

}

else {

if (ip[i] == '.') {

if (stoi(buff) > 255) return false;

if (numbers == 0) return false;

points++;

buff.clear();

numbers = 0;

if (ip[i + 1] == '.') return false;

}

else return false;

}

if (numbers > 3) return false;

if (points > 3) return false;

}

return true;

}

int main()

{

setlocale(LC\_ALL, "RUS");

unsigned char mac[6];

struct in\_addr srcip = { 0 };

struct sockaddr\_in sa;

char ip\_address[32];

bool flag = true;

WSADATA firstsock;

if (WSAStartup(MAKEWORD(2, 2), &firstsock) != 0)

{

cout << "Ошибка инициализации winsock";

cout << WSAGetLastError();

return -1;

}

do

{

if (!flag) cout << "Wrong address" << endl;

cout << "Enter IP: ";

cin >> ip\_address;

} while (!(flag = CheckAddress(ip\_address)));

//преобразование IP адреса другим способом

//srcip.s\_addr = inet\_addr(ip\_address);

inet\_pton(AF\_INET, ip\_address, &(sa.sin\_addr));

//Получение MAC по IP

GetMacAddress(mac, sa.sin\_addr);

//GetMacAddress(mac, srcip);

printf("MAC адрес : %.2X-%.2X-%.2X-%.2X-%.2X-%.2X",

mac[0], mac[1], mac[2], mac[3], mac[4], mac[5]);

printf("\n");

//\_getch();

return 0;

}

void GetMacAddress(unsigned char\* mac, struct in\_addr

destip)

{

DWORD ret;

IPAddr srcip;

ULONG MacAddr[2];

ULONG PhyAddrLen = 6;

int i;

srcip = 0;

//Послать ARP пакет

ret = SendARP((IPAddr)destip.S\_un.S\_addr,

srcip, MacAddr, &PhyAddrLen);

//Преобразовать адрес

if (PhyAddrLen)

{

BYTE\* bMacAddr = (BYTE\*)&MacAddr;

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

{

mac[i] = (char)bMacAddr[i];

}

}

}

Задача 19

Разработать программу, определяющую MAC-адреса всех доступных компьютеров сети из диапазона 172.16.192.100 – 172.16.193.10. Работу программы продемонстрировать на примере определения MAC-адресов двух компьютеров с IP-адресами 172.16.192.201 и 172.16.193.1.

Задача 20

Разработать программу, определяющую MAC-адреса всех доступных компьютеров сети из диапазона 172.16.192.100 – 172.17.193.10. Работу программы продемонстрировать на примере определения MAC-адресов двух компьютеров с IP-адресами 172.16.192.201 и 172.17.193.1.

Задача 21

Разработать программу, определяющую символьное имя компьютера по IP-адресу (IP-адрес ввести в диалоговом режиме). Для обработки исключительной ситуации – отсутствии в сети компьютера с введенным адресом – использовать процедуру try … catch.

ВАНЯ ЧТО-ТО НАКАЛЯКАЛ

(string name, string ip)[] bas= { ("ont", "192.16.172.1"), ("two", "192.16.172.2"), ("three", "192.16.172.3"), ("four", "192.16.172.4"), ("five", "192.16.172.5") };

Задача 22

Разработать программу, определяющую IP-адрес компьютера по символьному адресу (ввести в диалоговом режиме). Для обработки исключительной ситуации – отсутствии в сети компьютера с введенным адресом – использовать процедуру try … catch.

Задача 23

Разработать программу, определяющую IP-адрес компьютера по символьному адресу (ввести в диалоговом режиме). Организовать проверку на существование в символьном адресе следующих запрещенных символов.

Задача 24

Разработать программу, определяющую IP-адрес компьютера по символьному адресу (ввести в диалоговом режиме). Организовать использование шаблона имени следующего типа – знак «?» означает один любой символ (нижний регистр) латинского алфавита. Также сделать допущение, что знак «?» используется в имени лишь один раз.

Задача 25

Разработать программу, вычисляющую по IP-адресу и маске сети (вводятся в диалоговом режиме) NETWORK ID и HOST ID. Предусмотреть невозможность использования в качестве IP-адреса, адреса-исключения.

ИЗ ТОГО ФАЙЛА(С++)

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

/\*Разработать программу, вычисляющую по IP-адресу и маске сети NETWORK ID и HOST ID.\*/

bool check(int x, int x1, int x2, int x3) {

int arr[8], arr2[8], arr3[8], arr4[8];

int flag = 0;

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

{

arr[i] = ((int)((x >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i < 7) {

if ((arr[i] == 1) && (arr[i + 1] == 0)) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

flag = flag + 1;

}

}

}

//

if (flag > 0) {

return 0;

}

else {

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

{

arr2[i] = ((int)((x1 >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i == 0) {

if (arr[0] == 0 && arr2[0]) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

}

else {

if (i < 7) {

if ((arr2[i] == 1) && (arr2[i + 1] == 0)) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

flag = flag + 1;

}

}

}

}

}

//3 октет

if (flag > 0) {

return 0;

}

else {

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

{

arr3[i] = ((int)((x2 >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i == 0) {

if (arr2[0] == 0 && arr3[0]) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

}

else {

if (i < 7) {

if ((arr3[i] == 1) && (arr3[i + 1] == 0)) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

flag = flag + 1;

}

}

}

}

}

//4 октет

if (flag > 0) {

return 0;

}

else {

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

{

arr4[i] = ((int)((x3 >> i) & 1));

}

for (int i = 0; i < 8; i++) {

if (i == 0) {

if (arr3[0] == 0 && arr4[0]) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

}

else {

if (i < 7) {

if ((arr4[i] == 1) && (arr4[i + 1] == 0)) {

cout << "ТЫ ЛОХ ДАЖЕ МАСКУ НЕПРАВИЛЬНУЮ НАПИСАЛ" << endl;

break;

flag = flag + 1;

}

}

}

}

}

}

}

if (flag > 0) {

return 0;

}

else {

return 1;

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

string str;

cout << "Введите маску" << endl;

cin >> str;

string str\_x0, str\_x1, str\_x2, str\_x3;

int flag = 0;

for (int i = 0; i < str.size(); i++) {

if (str[i] == '.') {

flag = flag + 1;

}

if (flag == 0) {

if (str[i] != '.') {

str\_x0 = str\_x0 + str[i];

}

}

if (flag == 1) {

if (str[i] != '.') {

str\_x1 = str\_x1 + str[i];

}

}

if (flag == 2) {

if (str[i] != '.') {

str\_x2 = str\_x2 + str[i];

}

}

if (flag == 3) {

if (str[i] != '.') {

str\_x3 = str\_x3 + str[i];

}

}

}

//переведем в целочисленный

int x0 = atoi(str\_x0.c\_str()), x1 = atoi(str\_x1.c\_str()), x2 = atoi(str\_x2.c\_str()), x3 = atoi(str\_x3.c\_str());

int ch = check(x0, x1, x2, x3);

if (ch == 1) {

//IP

string str1;

cout << "Введите IP" << endl;

cin >> str1;

//сравнение полученного Ip с исключающим ip

if (str1 == "192.11.11.11")

{

cout << "IP адресс входит в адреса-исключения";

goto skip;

}

string str\_y0, str\_y1, str\_y2, str\_y3;

flag = 0;

for (int i = 0; i < str1.size(); i++) {

if (str1[i] == '.') {

flag = flag + 1;

}

if (flag == 0) {

if (str1[i] != '.') {

str\_y0 = str\_y0 + str1[i];

}

}

if (flag == 1) {

if (str1[i] != '.') {

str\_y1 = str\_y1 + str1[i];

}

}

if (flag == 2) {

if (str1[i] != '.') {

str\_y2 = str\_y2 + str1[i];

}

}

if (flag == 3) {

if (str1[i] != '.') {

str\_y3 = str\_y3 + str1[i];

}

}

}

//переведем в целочисленный

int y0 = atoi(str\_y0.c\_str()), y1 = atoi(str\_y1.c\_str()), y2 = atoi(str\_y2.c\_str()), y3 = atoi(str\_y3.c\_str());

cout << "NETWORK ID" << endl;

cout << (y0 & x0) << ".";

cout << (y1 & x1) << ".";

cout << (y2 & x2) << ".";

cout << (y3 & x3) << endl;

cout << "HOST ID" << endl;

x0 = ~x0; x1 = ~x1; x2 = ~x2; x3 = ~x3;

cout << (y0 | x0) \* (-1) << ".";

cout << (y1 | x1) \* (-1) << ".";

cout << (y2 | x2) \* (-1) << ".";

cout << (y3 | x3) \* (-1) << endl;

}

else {

cout << "Маска неправильная перезапусти прогу дурак" << endl;

}

skip:

cout << endl;

}