**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

***Департамент анализа больших данных и машинного обучения***

Дисциплина «Организация вычислительных систем»

**Практическая работа №1**

**«Работа с IP адресами»**

**Выполнил:**

Обучающийся группы ПИ20-2

Зайцев Никита Валерьевич

1 курс, 2 семестр

**Проверил**:

к.т.н., доцент

Петросов Давид Арегович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва 2021**

**Задание №1.** Выполнить ручной перевод IP адресов, представленных десятичной записью в двоичную, и наоборот:

1) 10.124.56.220 → 00001010.01111100.00111000.11011100

2) 113.72.101.11 →01110001.01001000.01100101.00001011

3) 173.143.32.194 →10101101.10001111.00100000.11000010

4) 200.69.139.217 →11001000.01000101.10001011.11011001

5) 88.212.236.76 → 01011000.11010100.11101100.01001100

6) 01011101.10111011.01001000.00110000 →93.187.72.48

7) 01001000.10100011.00000100.10100001 →72.163.4.161

8) 00001111.11011001.11101000.11110101 →15.217.232.245

9) 01000101.00010100.00111011.01010000 →69.20.59.80

10) 00101011.11110011.10000010.00111101 →43.243.130.61

**Задание №2**

**Изображение выглядит как текст, кроссворд

Автоматически созданное описание**

1) Записать маску для проекта: сеть 172.16.0.0. 250 подсетей и 220 хостов.

Сеть класса В

16 бит занято под сеть (N=16)

16 бит свободно

Необходимо 250 подсетей, соответственно берем 8 бит под сети (256>250) (S=8)

16 – 8 = 8 бит под хосты (256>220) (H=8)

16 + 8 + 8 = 32

N + S = 24

Соответственно получаем маску 255.255.255.0

2) Записать маску для проекта: сеть 10.0.0.0. 2000 подсетей и 1500 хостов

Сеть класса А

8 бит занято (N=8)

24 бит свободно

Под подсети нужно 11 бит (2048>2000) (S=11), соответственно под хосты остается 24-11 = 13 бит (H=13)

N + S = 19

Соответственно получаем маску 255.255.224.0

3) Записать маску для проекта: сеть 192.168.0.0. 4 подсети и 60 хостов.

хостов.

Сеть класса C

24 бит занято под сеть (N=24)

8 бит свободно

Под подсети нужно 2 бит (4>4) (S = 3), под хосты свободно 6 бит

N + S = 26

Соответственно получаем маску 255.255.255.192

**Задание №4** Реализовать программное средство для представления IP адресов с десятичного и двоичного кодирования. В качестве входных данных задается десятичное представление IP адреса, программное средство выдает двоичное представление или пользователь может ввести двоичное представление IP адреса и получить десятичное представление.

*print*('Введите IP адрес')  
ip = *input*().split('.')  
*print*(ip)  
ans = ''  
*if len*(ip[0]) == 8:  
 *for* i *in* ip:  
 ans += *str*(*int*(i, 2)) + '.'  
*else*:  
 *for* i *in* ip:  
 b = *str*(*bin*(*int*(i))[2:]) + '.'  
 *if len*(b) < 8:  
 b = '0' \* (8 - *len*(b)) + b  
 ans += b  
  
ans = ans[0:*len*(ans) - 1]  
*print*(ans)

Скриншот:

Изображение выглядит как текст

Автоматически созданное описание

**Задание № 5** Реализовать программное средство, способное определять класс сети, начальный и конечный адрес и маску подсети, на основании заданного IP адреса.

*print*('Введите IP адрес')  
ip = *input*().split('.')  
b = *str*(*bin*(*int*(ip[0]))[2:])  
*if len*(b) < 8:  
 b = '0' \* (8 - *len*(b)) + b  
b = b[0:4]  
*if* b[0] == '0':  
 *print*('Класс сети A, начальный адрес 0.0.0.0, конечный адрес 127.255.255.255, маска 255.0.0.0')  
*elif* b[0:2] == '10':  
 *print*('Класс сети B, начальный адрес 128.0.0.0, конечный адрес 191.255.255.255, маска 255.255.0.0')  
*elif* b == '1100':  
 *print*('Класс сети C, начальный адрес 192.0.0.0, конечный адрес 223.255.255.255, маска 255.255.255.0')  
*elif* b == '1110':  
 *print*('Класс сети D, начальный адрес 224.0.0.0, конечный адрес 239.255.255.255, маска 255.255.255.0')  
*elif* b == '1111':  
 *print*('Класс сети E, начальный адрес 240.0.0.0, конечный адрес 255.255.255.255, маска 255.255.255.0')

Скриншот:

Изображение выглядит как текст

Автоматически созданное описание

**Задание № 6** Разработать программное средство, которое по заданной сети, количеству подсетей и хостов формирует: маску для проекта сети, а также определяет класс сетей, начало и конец сети, количество IP-адресов в сети, количество доступных IP-адресов в сети для назначения хостам, стек первых 5 допустимых IP-адресов, стек последних 5 допустимых IP-адресов.

*print*('Введите IP, кол-во подсетей, и кол-во хостов\nв формате <xxx.xxx.xxx.xxx,Y,Z>')  
ip, subnet, hosts = *input*().split(',')  
subnet = *int*(subnet)  
hosts = *int*(hosts)  
ip = ip.split('.')  
ip\_class = ''  
bits\_occupied = 0  
bits\_free = 0  
bit = *str*(*bin*(*int*(ip[0]))[2:])  
*if len*(bit) < 8:  
 b = '0' \* (8 - *len*(bit)) + bit  
bit = bit[0:4]  
  
*if* bit[0] == '0':  
 ip\_class = 'A'  
 bits\_occupied = 8  
*elif* bit[0:2] == '10':  
 ip\_class = 'B'  
 bits\_occupied = 16  
*elif* bit == '1100':  
 ip\_class = 'C'  
 bits\_occupied = 24  
bits\_free = 32 - bits\_occupied  
*print*(bits\_free, ip\_class)  
bits\_subnet = 0  
  
*for* i *in range*(0, 99, 2):  
 *if* 2 \*\* i >= subnet:  
 bits\_subnet = i  
 *print*(i)  
 *break*bits\_hosts = bits\_free - bits\_subnet  
mask\_bits = bits\_occupied + bits\_hosts  
*print*(mask\_bits)  
mask = ''  
mask\_10 = ''  
  
*for* i *in range*(1, 33):  
 *if* mask\_bits == 0:  
 mask += '0'  
 *else*:  
 mask += '1'  
 mask\_bits -= 1  
 *if* i != 0 *and* i % 8 == 0:  
 mask += '.'  
*print*(mask)  
  
mask\_pie = ''  
*for* i *in* mask:  
 *if* i == '.':  
 mask\_10 += f'{*int*(mask\_pie, 2)}.'  
 mask\_pie = ''  
 *continue* mask\_pie += i  
  
*print*(mask\_10)  
kolvooktet = mask\_bits // 8  
*print*("Количество возможных сетей = ", 2 \*\* bits\_occupied)  
*if* kolvooktet <= 3:  
 *print*("Количество доступных сетей = ", 2 \*\* bits\_occupied - 2)  
*else*:  
 *print*("Количество доступных сетей = ", 2 \*\* bits\_occupied - 1)  
*print*("Стек первых 5 допустимых IP-адресов :")  
*if* kolvooktet == 3:  
 *for* i *in range*(1, 6):  
 *print*(*str*(ip[0]) + "." + (*str*(ip[1]) + "." + (*str*(ip[2]) + "." + *str*(i))))  
*if* kolvooktet == 2:  
 *for* i *in range*(1, 6):  
 *print*(*str*(ip[0]) + "." + (*str*(ip[1]) + "." + *str*(*int*(ip[2]) & *int*(p)) + "." + *str*(i)))  
*elif* kolvooktet == 1:  
 *for* i *in range*(1, 6):  
 *print*(*str*(ip[0]) + "." + (*int*(ip[1]) & *int*(p)) + "." + "0" + "." + *str*(i))  
*print*("Стек последних 5 допустимых IP-адресов :")  
*if* kolvooktet == 3:  
 *for* i *in range*((24 - mask\_bits) - 1, -1, -1):  
 e = (2 \*\* i) + e  
 *for* i *in range*(1, 6):  
 *print*(*str*(ip[0]) + "." + *str*(ip[1]) + "." + *str*(ip[2]) + "." + *str*(249 + i))  
*if* kolvooktet == 2:  
 *for* i *in range*((24 - mask\_bits) - 1, -1, -1):  
 e = (2 \*\* i) + e  
 *for* i *in range*(1, 6):  
 *print*(*str*(ip[0]) + "." + *str*(ip[1]) + "." + *str*(*int*(ip[2]) & *int*(p) | *int*(e)) + "." + *str*(  
 249 + i))  
*elif* kolvooktet == 1:  
 *for* i *in range*((16 - mask\_bits) - 1, -1, -1):  
 e = (2 \*\* i) + e  
 *for* i *in range*(1, 6):  
 *print*(*str*(ip[0]) + "." + *str*(*int*(ip[1]) & *int*(p)) + "." + *str*(  
 *int*(ip[2]) & *int*(p) | *int*(e)) + +"." + *str*(249 + i))  
  
*# 172.16.0.0,250,220 \\ 10.0.0.0,2000,1500*

Скриншот:

Изображение выглядит как текст

Автоматически созданное описание