Федеральное государственное образовательное бюджетное учреждение  
высшего образования  
«Финансовый университет при Правительстве РФ»

Отчет по практике №1. Работа с IP адресами

Выполнил студент:

Глущенко Никита Андреевич Группы: ПИ20-2

Преподаватель:

Петросов Давид Арегович

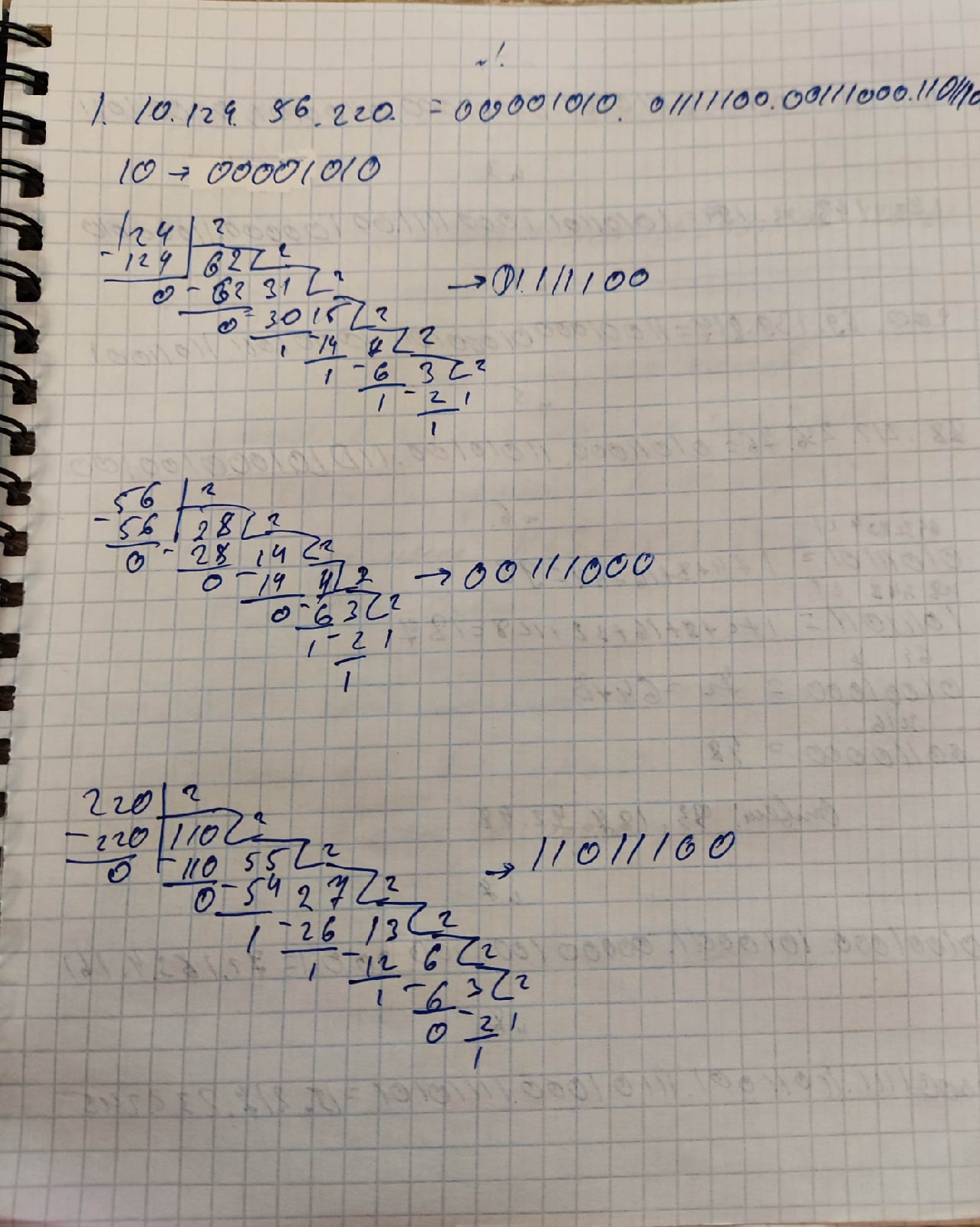
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

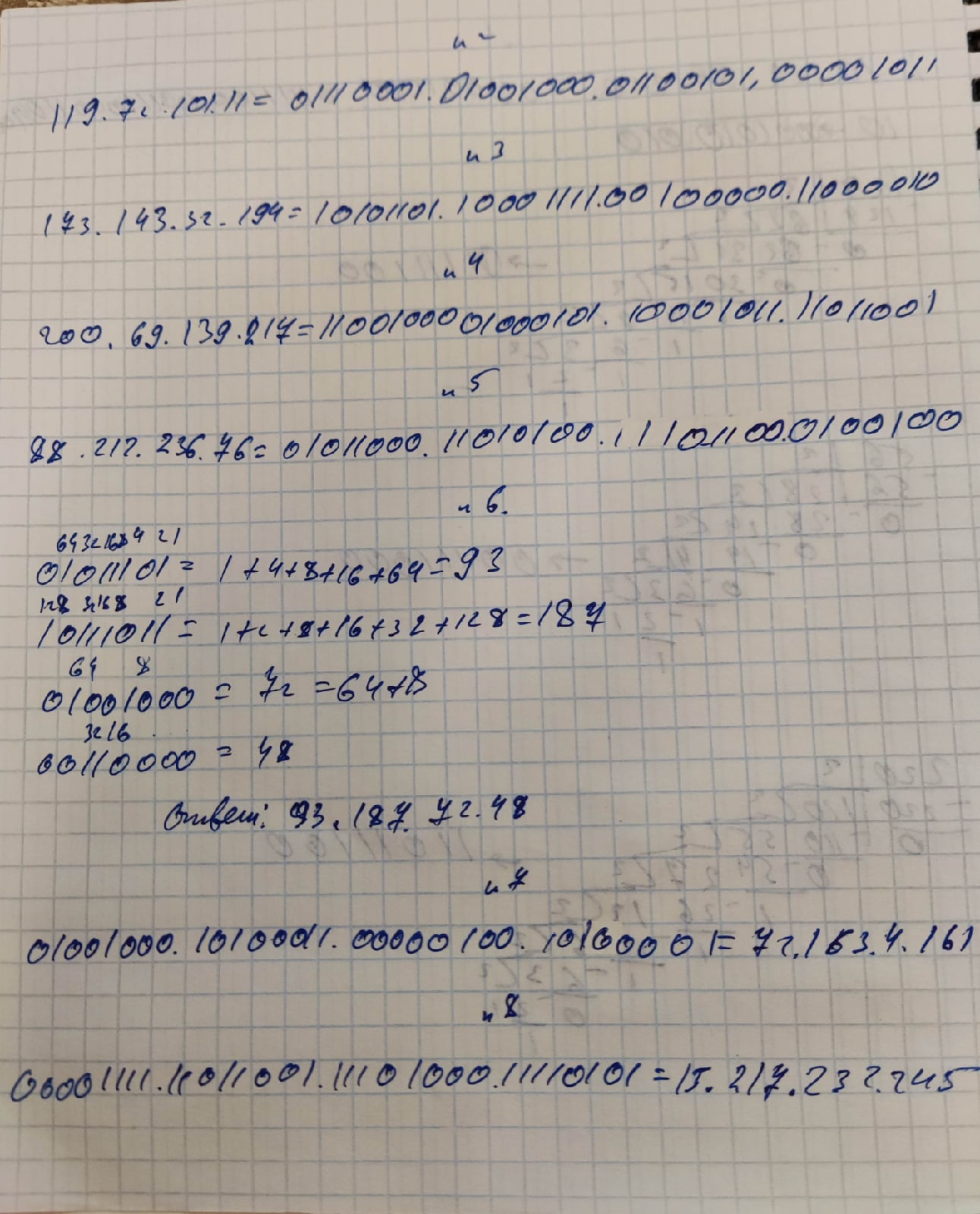
(подпись)

Москва 2021

Задание №1.

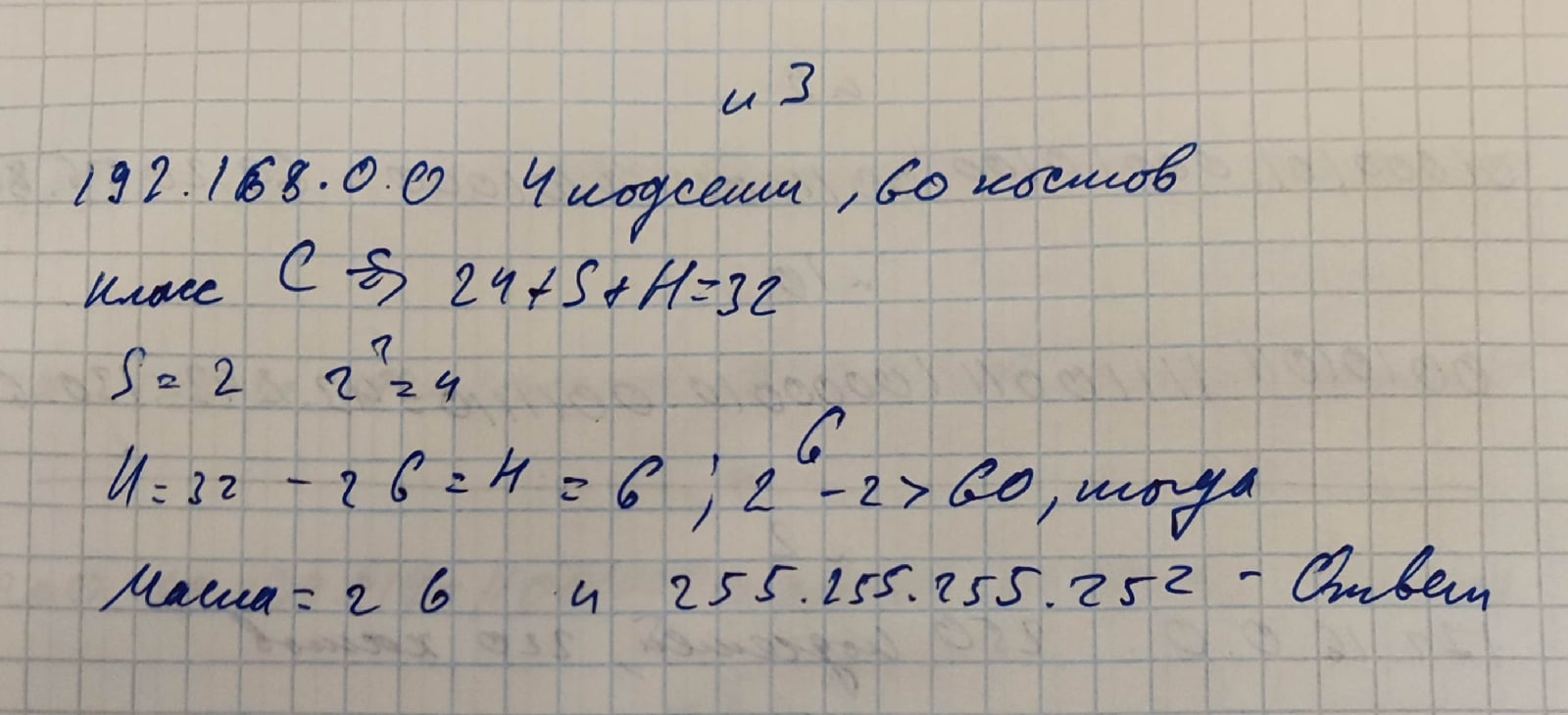
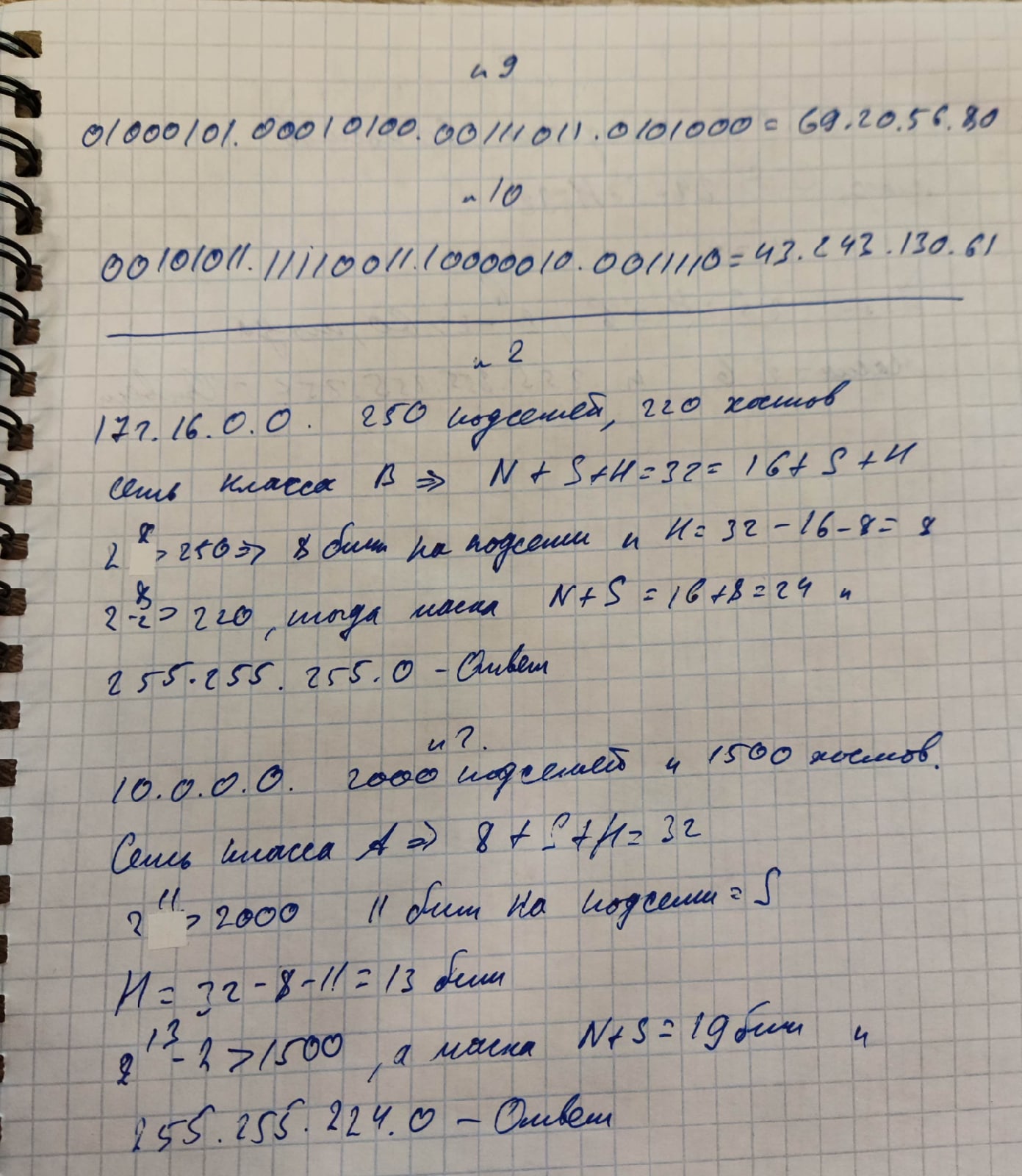
Выполнить ручной перевод IP адресов, представленных десятичной записью в двоичную, и наоборот:





Задание №2.

Записать маски для проектов:



Задание №3.

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. На рисунке 1 представлен перевод двоичного формата записи адреса в десятичный, а на рисунке 2 из десятичного в двоичный.

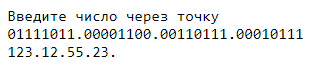


Рисунок 1 - перевод двоичного формата записи адреса в десятичный

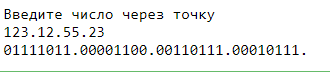


Рисунок 2 - перевод десятичного формата записи адреса в двоичный

Листинг программы:

print('Введите число через точку')

ipAdress = input().split('.')

buf = ''

if len(ipAdress[0]) == 8:

for i in range(4):

buf = int(ipAdress[i], 2)

print(buf, end='.')

else:

for i in range(4):

while int(ipAdress[i]) > 0:

buf = str(int(ipAdress[i]) % 2) + buf

ipAdress[i] = int(ipAdress[i]) // 2

if len(buf) < 8:

buf = '0'\*(8-len(buf)) + buf

print(buf, end = '.')

buf = ''

Задание №4

Реализовать программное средство, способное определять класс сети, начальный и конечный адрес и маску подсети, на основании заданного IP адреса. На рисунке 3 представлено определение класса сети через двоичный формат записи адреса, а на рисунке 4 через десятичный.

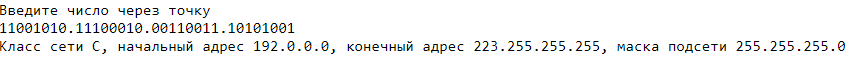


Рисунок 3 - определение класса сети через двоичный формат записи адреса

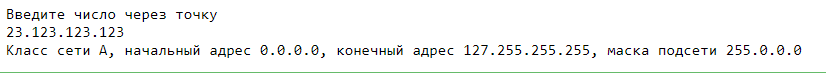


Рисунок 4 - определение класса сети через десятичный формат записи адреса

Листинг:

print('Введите число через точку')

ipAdress = input().split('.')

if len(ipAdress[0]) == 8:

ipAdress[0] = int(ipAdress[0], 2)

if ipAdress[0]>=0 and ipAdress[0]<128:

print('Класс сети A, начальный адрес 0.0.0.0, конечный адрес 127.255.255.255, маска подсети 255.0.0.0')

elif ipAdress[0]>=128 and ipAdress[0]<192:

print('Класс сети B, начальный адрес 128.0.0.0, конечный адрес 191.255.255.255, маска подсети 255.255.0.0')

elif ipAdress[0]>=192 and ipAdress[0]<224:

print('Класс сети C, начальный адрес 192.0.0.0, конечный адрес 223.255.255.255, маска подсети 255.255.255.0')

elif ipAdress[0]>=224 and ipAdress[0]<240:

print('Класс сети D, начальный адрес 224.0.0.0, конечный адрес 239.255.255.255, маска подсети 255.255.255.0')

elif ipAdress[0]>=240 and ipAdress[0]<256:

print('Класс сети E, начальный адрес 240.0.0.0, конечный адрес 255.255.255.255, маска подсети 255.255.255.0')

else:

ipAdress[0] = int(ipAdress[0])

if ipAdress[0]>=0 and ipAdress[0]<128:

print('Класс сети A, начальный адрес 0.0.0.0, конечный адрес 127.255.255.255, маска подсети 255.0.0.0')

elif ipAdress[0]>=128 and ipAdress[0]<192:

print('Класс сети B, начальный адрес 128.0.0.0, конечный адрес 191.255.255.255, маска подсети 255.255.0.0')

elif ipAdress[0]>=192 and ipAdress[0]<224:

print('Класс сети C, начальный адрес 192.0.0.0, конечный адрес 223.255.255.255, маска подсети 255.255.255.0')

elif ipAdress[0]>=224 and ipAdress[0]<240:

print('Класс сети D, начальный адрес 224.0.0.0, конечный адрес 239.255.255.255, маска подсети 255.255.255.0')

elif ipAdress[0]>=240 and ipAdress[0]<256:

print('Класс сети E, начальный адрес 240.0.0.0, конечный адрес 255.255.255.255, маска подсети 255.255.255.0')

Задание № 5

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.

На рисунках 5,6,7 представлены решения задачи и проверка через онлайн-калькулятор.

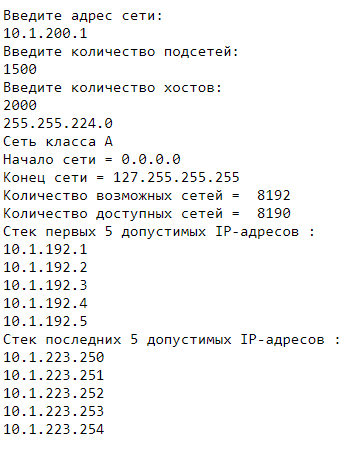
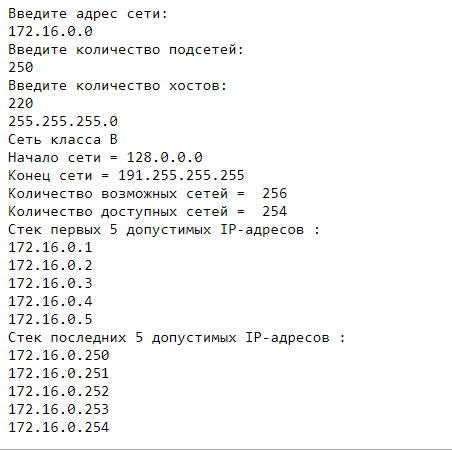


Рисунок 5 – решение задачи



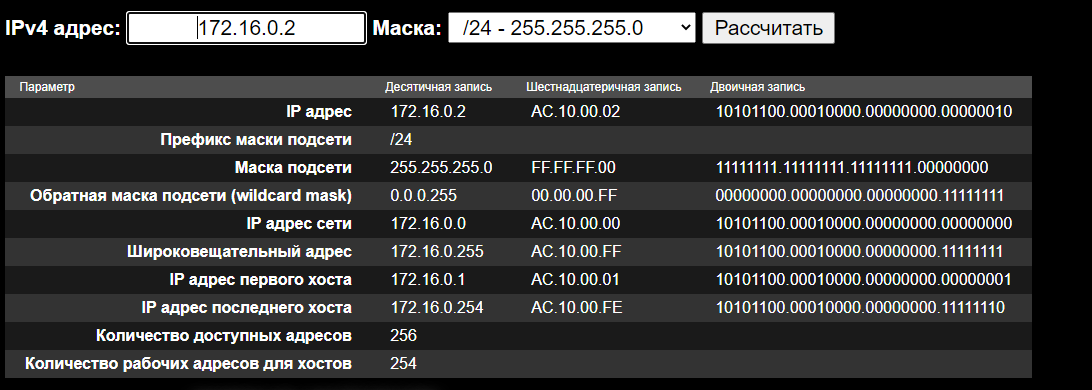


Рисунок 6 – решение задачи

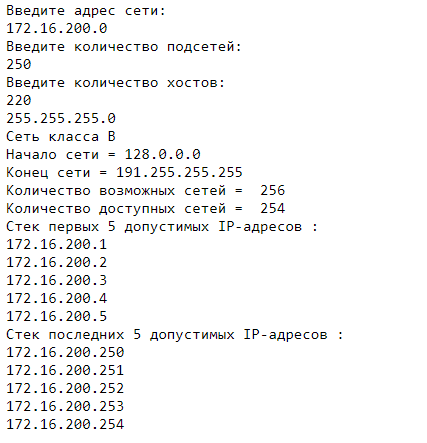
 

Рисунок 7 – решение задачи

Листинг:

print('Введите адрес сети:')

ipAdress = input().split('.')

print('Введите количество подсетей:')

kolvopodset = int(input())

print('Введите количество хостов:')

kolvohost = int(input())

N = 0 #количество битов сети по классу

b = 0 #S в формуле

mask = 0

formask = 0

H = 0

kolvooktet = 0

mask2 = 0

p = 0

p1 = 0

k = 0

e=0

mask = ''

ipAdress[0] = int(ipAdress[0])

if (ipAdress[0] > 0) and (ipAdress[0] < 128):

N = 8

while (2\*\*b - 2) < kolvopodset:

b +=1

else:

H = 32 - (N+b)

if 2\*\*H >= kolvohost:

formask = N+b

kolvooktet = formask // 8

mask2 = formask % 8

if mask2 != 0:

for i in range(7, -1, -1):

p = (2\*\*i)+p

mask2 = mask2 - 1

if mask2 <= 0:

break

p = str(p)

k = (32-(N+b))

l = k // 8 -1

for i in range(0, l+2):

p1 = '.'\*i + '0'\* i

mask = ('255.')\*kolvooktet + p + (p1)

print(mask)

print('Сеть класса A')

print("Начало сети = 0.0.0.0")

print("Конец сети = 127.255.255.255")

print("Количество возможных сетей = ", 2\*\*H)

if kolvooktet<=4:

print("Количество доступных сетей = ", 2\*\*H - 2)

else:

print("Количество доступных сетей = ", 2\*\*H - 1)

print("Стек первых 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+(str(ipAdress[2])+"."+str(i))))

if kolvooktet==2:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p))+"."+str(i)))

elif kolvooktet==1:

for i in range(1,6):

print(str(ipAdress[0])+"."+(int(ipAdress[1]) & int(p))+"."+"0"+"."+str(i))

print("Стек последних 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(ipAdress[2])+"."+str(249+i))

if kolvooktet==2:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p) | int(e))+"."+str(249+i))

elif kolvooktet==1:

for i in range((16-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(int(ipAdress[1]) & int(p))+"."+str(int(ipAdress[2]) & int(p) | int(e))++"."+str(249+i))

elif (ipAdress[0] > 127) and (ipAdress[0] < 192):

N = 16

while (2\*\*b - 2) < kolvopodset:

b +=1

else:

H = 32 - (N+b)

if 2\*\*H >= kolvohost:

formask = N+b

kolvooktet = formask // 8

mask2 = formask % 8

if mask2 != 0:

for i in range(7, -1, -1):

p = (2\*\*i)+p

mask2 = mask2 - 1

if mask2 <= 0:

break

p = str(p)

k = (32-(N+b))

l = k // 8 -1

for i in range(0, l+1):

p1 = '.'\*i + '0'\* i

mask = ('255.')\*kolvooktet + p + p1

print(mask)

print('Сеть класса B')

print("Начало сети = 128.0.0.0")

print("Конец сети = 191.255.255.255")

print("Количество возможных сетей = ", 2\*\*H)

if kolvooktet<=4:

print("Количество доступных сетей = ", 2\*\*H - 2)

else:

print("Количество доступных сетей = ", 2\*\*H - 1)

print("Стек первых 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+(str(ipAdress[2])+"."+str(i))))

if kolvooktet==2:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p))+"."+str(i)))

elif kolvooktet==1:

for i in range(1,6):

print(str(ipAdress[0])+"."+(int(ipAdress[1]) & int(p))+"."+"0"+"."+str(i))

print("Стек последних 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(ipAdress[2])+"."+str(249+i))

if kolvooktet==2:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p) | int(e))+"."+str(249+i))

elif kolvooktet==1:

for i in range((16-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(int(ipAdress[1]) & int(p))+"."+str(int(ipAdress[2]) & int(p) | int(e))++"."+str(249+i))

elif (ipAdress[0] > 191) and (ipAdress[0] < 224):

N = 24

while (2\*\*b - 2) < kolvopodset:

b +=1

else:

H = 32 - (N+b)

if 2\*\*H >= kolvohost:

formask = N+b

kolvooktet = formask // 8

mask2 = formask % 8

if mask2 != 0:

for i in range(7, -1, -1):

p = (2\*\*i)+p

mask2 = mask2 - 1

if mask2 <= 0:

break

p = str(p)

k = (32-(N+b))

l = k // 8

for i in range(0, l+1):

p1 = '.'\*i + '0'\* i

mask = ('255.')\*kolvooktet + p + p1

print(mask)

print('Сеть класса С')

print("Начало сети = 192.0.0.0")

print("Конец сети = 223.255.255.255")

print("Количество возможных сетей = ", 2\*\*H)

if kolvooktet<=3:

print("Количество доступных сетей = ", 2\*\*H - 2)

else:

print("Количество доступных сетей = ", 2\*\*H - 1)

print("Стек первых 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+(str(ipAdress[2])+"."+str(i))))

if kolvooktet==2:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p))+"."+str(i)))

elif kolvooktet==1:

for i in range(1,6):

print(str(ipAdress[0])+"."+(int(ipAdress[1]) & int(p))+"."+"0"+"."+str(i))

print("Стек последних 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(ipAdress[2])+"."+str(249+i))

if kolvooktet==2:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p) | int(e))+"."+str(249+i))

elif kolvooktet==1:

for i in range((16-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(int(ipAdress[1]) & int(p))+"."+str(int(ipAdress[2]) & int(p) | int(e))++"."+str(249+i))

elif (ipAdress[0] > 223) and (ipAdress[0] < 240):

N = 28

while (2\*\*b - 2) < kolvopodset:

b +=1

else:

H = 32 - (N+b)

if 2\*\*H >= kolvohost:

formask = N+b

kolvooktet = formask // 8

mask2 = formask % 8

if mask2 != 0:

for i in range(7, -1, -1):

p = (2\*\*i)+p

mask2 = mask2 - 1

if mask2 <= 0:

break

p = str(p)

k = (32-(N+b))

l = k // 8 -1

for i in range(0, l+1):

p1 = '.'\*i + '0'\* i

mask = ('255.')\*kolvooktet + p + p1

print(mask)

print('Сеть класса D')

print("Начало сети = 224.0.0.0")

print("Конец сети = 239.255.255.255")

print("Количество возможных сетей = ", 2\*\*H)

if kolvooktet<=3:

print("Количество доступных сетей = ", 2\*\*H - 2)

else:

print("Количество доступных сетей = ", 2\*\*H - 1)

print("Стек первых 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+(str(ipAdress[2])+"."+str(i))))

if kolvooktet==2:

for i in range(1,6):

print(str(ipAdress[0])+"."+(str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p))+"."+str(i)))

elif kolvooktet==1:

for i in range(1,6):

print(str(ipAdress[0])+"."+(int(ipAdress[1]) & int(p))+"."+"0"+"."+str(i))

print("Стек последних 5 допустимых IP-адресов :")

if kolvooktet==3:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(ipAdress[2])+"."+str(249+i))

if kolvooktet==2:

for i in range((24-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(ipAdress[1])+"."+str(int(ipAdress[2]) & int(p) | int(e))+"."+str(249+i))

elif kolvooktet==1:

for i in range((16-formask)-1, -1, -1):

e = (2\*\*i)+e

for i in range(1,6):

print(str(ipAdress[0])+"."+str(int(ipAdress[1]) & int(p))+"."+str(int(ipAdress[2]) & int(p) | int(e))++"."+str(249+i))

print('Сеть класса D')

Выводы: по результатам практики 1 познакомился с понятием ip-адресов и их масок в целом, а также научился определять маски для проектов. Узнал, какие классы бывают у ip-адресов и как их отличать друг от друга. Смог реализовать программные средства для представления IP адресов с десятичного и двоичного кодирования, для определения классов и для построения масок. Также я поработал с начальными и конечными адресами и стеками допустимых ip-адресов в сети.