

Рабочий лист по теме  
«IP-адресация: аналитика и Python»  
(задание №13 КЕГЭ)

## Часть 1. Аналитическое решение

### Теория (кратко)

- **IP-адрес** – 32-битное число, записываемое четырьмя октетами (например, 192.168.1.10).
- **Маска подсети** – тоже 32 бита, где сначала идут единицы (сеть), затем нули (узлы). Пример: 255.255.255.0 → 24 единицы, 8 нулей.
- **Адрес сети** = поразрядное AND между IP-адресом узла и маской.
- **Широковещательный адрес** – все биты, соответствующие нулям в маске, равны 1.
- **Количество адресов в сети** = 2.
- **Доступные адреса для компьютеров** – все, кроме адреса сети и широковещательного.

*Важно:* при ручном вычислении полезно переводить октеты в двоичную систему (8 бит) и выполнять поразрядные операции.

### Задания для самостоятельного решения

**Задание 1.** IP-адрес узла – 192.168.32.160, маска – 255.255.255.240.

Найдите адрес сети. Ответ запишите в виде четырёх десятичных чисел через точку.

**Задание 2.** Для узла с IP-адресом 172.16.88.45 адрес сети равен 172.16.88.32. Чему равно **наименьшее возможное** значение четвёртого байта маски? Ответ дайте в виде десятичного числа.

**Задание 3.** IP-адрес узла – 10.120.45.67, маска – 255.255.252.0. Найдите **наибольший** IP-адрес, который можно назначить компьютеру в этой сети. Ответ укажите в виде **суммы** числовых значений его октетов.

**Задание 4.** Маска подсети – 255.254.0.0. Сколько всего IP-адресов (включая адрес сети и широковещательный) принадлежит этой сети?

**Задание 5.** Узел имеет IP-адрес 215.181.200.27. Известно, что адрес сети равен 215.181.192.0.

Определите маску подсети (в десятичном виде, все четыре октета).

*Указание: рассмотрите третий байт.*

## Часть 2. Программное решение на Python

### Теория (основные функции и приёмы)

Для работы с IP-адресами в Python используется встроенный модуль `ipaddress`. Обязательный импорт:

```
from ipaddress import *
```

Что делаем	Как пишем
Создать объект сети по адресу узла	<code>net = ip_network('host/mask', 0)</code> (параметр 0 или <code>strict=False</code> обязателен)
Получить десятичную маску	<code>net.netmask</code>
Получить количество адресов (включая сетевой и широковещательный)	<code>net.num_addresses</code>
Адрес сети	<code>net[0]</code>
Широковещательный адрес	<code>net[-1]</code>
Первый доступный адрес для ПК	<code>net[1]</code>
Последний доступный адрес для ПК	<code>net[-2]</code>
Перебрать все адреса сети	<code>for ip in net:</code>
Преобразовать строку в объект IPv4Address	<code>ip_address('192.168.1.1')</code>
Перевести IP-адрес в 32-битный двоичный вид	<code>f{int(ip):032b}</code>
Перевести адрес сети в двоичный вид	<code>f{int(net[0]):032b}</code>
Получить сумму октетов IP	<code>sum(int(x) for x in str(ip).split('.'))</code>
Удалить точки из строки IP	<code>str(ip).replace('.', '')</code>

### Задания на знание и применение функций

#### Задание 6. (соотнесение)

Соедините линиями выражение из левого столбца с его описанием из правого.

Выражение	Описание
1. <code>ip_network('10.0.0.1/24', 0)</code>	А. Возвращает десятичную маску сети
2. <code>net.netmask</code>	Б. Количество адресов в сети (включая все)
3. <code>net.num_addresses</code>	В. Создаёт объект сети по адресу узла
4. <code>ip_address('192.168.0.1')</code>	Г. Адрес сети (первый адрес)
5. <code>net[0]</code>	Д. Широковещательный адрес
6. <code>net[-1]</code>	Е. Создаёт объект отдельного IP-адреса
7. <code>f{int(ip):032b}</code>	Ж. Двоичная строка длиной 32 бита

**Задание 7.** (выбор правильных вариантов)

Отметьте **все** корректные способы решить указанную задачу.

**А.** Получить общее количество адресов для сети, заданной узлом 10.20.30.40 и маской 255.255.255.240.

1. `ip_network('10.20.30.40/255.255.255.240', 0).num_addresses`
2. `ip_address('10.20.30.40').num_addresses`
3. `len(ip_network('10.20.30.40/255.255.255.240', 0))`
4. `ip_network('10.20.30.40/255.255.255.240', 0).num_addresses`

**Б.** Получить **наибольший** IP-адрес, который можно назначить компьютеру в сети 192.168.0.0/24 (задана узлом 192.168.0.5).

1. `ip_network('192.168.0.5/24', 0)[-2]`
2. `ip_network('192.168.0.5/24', 0)[-1]`
3. `ip_network('192.168.0.5/24', 0)[1]`
4. `ip_network('192.168.0.5/24', 0)[0]`

**В.** Перевести объект IPv4Address (например, `ip = ip_address('172.16.1.1')`) в 32-битную двоичную строку с ведущими нулями.

1. `bin(ip)`
2. `f{int(ip):032b}`
3. `ip.binary()`
4. `format(int(ip), '032b')`

**Задание 8.** (что выведет программа?)

Напишите, что будет напечатано. Если возникнет ошибка – укажите «ошибка» и объясните почему.

**Фрагмент 1**

```
from ipaddress import *
net = ip_network('10.0.0.0/255.255.255.0', 0)
print(net.netmask)
print(net.num_addresses)
```

Ответ: \_\_\_\_\_

**Фрагмент 2**

```
from ipaddress import *
ip = ip_address('192.168.1.100')
print(f'{int(ip):032b}')
print(str(ip).replace('.', ''))
```

Ответ: \_\_\_\_\_

**Фрагмент 3**

```
from ipaddress import *
net = ip_network('172.16.5.37/255.255.255.192', 0)
print(net[0])
print(net[-1])
print(net[-2])
```

Ответ: \_\_\_\_\_

**Фрагмент 4**

```
from ipaddress import *
net = ip_network('8.8.8.8/29', 0)
cnt = 0
for ip in net:
    if ip == net[0] or ip == net[-1]:
        continue
    cnt += 1
print(cnt)
```

Ответ: \_\_\_\_\_

**Задание 9.** (найди ошибку)

В каждом фрагменте кода есть ошибка (одна или несколько). Найдите их и напишите исправленный вариант.

**А.**

```
net = ip_network('192.168.0.10 255.255.255.0')
print(net[0])
```

Исправление:

**Б.**

```
from ipaddress import *
ip = ip_address('10.10.10.10')
bin_ip = f'{ip:b}'
print(bin_ip)
```

Исправление:

**В.**

```
from ipaddress import *
net = ip_network('10.0.0.1/24', 0)
print(net.netmask)
print(net.num_addresses) #
```

Исправление:

**Задание 10.** (написание кода)

Напишите программы, решающие следующие задачи. Код должен быть полным (с импортом) и выводить только требуемый результат.

**А.** Для сети, заданной узлом 185.49.83.72 и маской 255.255.240.0, выведите:

- десятичную маску (используя свойство `netmask`);
- общее количество адресов (используя `num_addresses`).

**Б.** Для сети 167.66.136.176/255.254.0.0 найдите **наименьший** IP-адрес, который можно назначить компьютеру, и выведите **сумму** его октетов.

**В.** Для сети 191.128.66.83/255.192.0.0 найдите **наибольший** доступный адрес и выведите его **без точек** (например, 191191255254).

**Г.** Для сети 10.20.30.40/255.255.255.128 подсчитайте, сколько IP-адресов, доступных для компьютеров, содержат в своей 32-битной двоичной записи **ровно две** единицы. Программа должна вывести это количество.

**Задание 11.** (комбинированное)

Используя аналитический метод, вы нашли маску для задачи 5 (где IP=215.181.200.27, адрес сети=215.181.192.0).

Теперь напишите программу, которая создаёт сеть по этому IP и **найденной вами маске**, а затем выводит адрес сети, полученный программно. Убедитесь, что он совпадает с 215.181.192.0.

Если совпало – программа должна вывести слово **ВЕРНО**, иначе – **НЕВЕРНО**.

# Ответы для самопроверки

## Часть 1. Аналитика

1. 192.168.32.144 (160 & 240 = 144)
2. 32 (четвёртый байт маски = 00100000<sub>2</sub>)
3. 431 (сеть 10.120.44.0, последний доступный 10.120.47.254 → сумма 10+120+47+254 = 431)
4. 131072 (2<sup>17</sup>, т.к. в маске 255.254.0.0 – 15 единиц, 17 нулей)
5. 255.255.224.0 (третий байт 224)

## Часть 2. Python

Задание 6 (соотнесение): 1–В, 2–А, 3–Б, 4–Е, 5–Г, 6–Д, 7–Ж.

Задание 7 (выбор):

- А) правильные: 1 и 4 (2 – неверно, 3 – `len` не работает)
- Б) правильный: 1 (индекс -2)
- В) правильные: 2 и 4

Задание 8 (что выведет):

1. Фрагмент 1: 255.255.255.0  
256
2. Фрагмент 2: 11000000101010000000000101100100  
1921681100
3. Фрагмент 3: 172.16.5.0  
172.16.5.63  
172.16.5.62
4. Фрагмент 4: 6

Задание 9 (исправление ошибок):

- А) Ошибки: нет импорта, разделитель – пробел вместо `'/'`.  
Исправление:

```
from ipaddress import *
net = ip_network('192.168.0.10/255.255.255.0', 0)
print(net[0])
```

- Б) Ошибка: `f{ip:b}` – нельзя применить к объекту. Надо `f{int(ip):032b}`.
- В) Опечатка: `num_adresses` → `num_addresses`.

Задание 10 (примеры решений – для учителя):

```

#
from ipaddress import *
net = ip_network('185.49.83.72/255.255.240.0', 0)
print(net.netmask)
print(net.num_addresses)

#
net = ip_network('167.66.136.176/255.254.0.0', 0)
ip = net[1]
print(sum(int(x) for x in str(ip).split('.')))

#
net = ip_network('191.128.66.83/255.192.0.0', 0)
ip = net[-2]
print(str(ip).replace('.', ''))

#
net = ip_network('10.20.30.40/255.255.255.128', 0)
cnt = 0
for ip in net:
    if ip == net[0] or ip == net[-1]:
        continue
    if f'{int(ip):032b}'.count('1') == 2:
        cnt += 1
print(cnt)

```

**Задание 11 (комбинированное):** Программа должна вывести ВЕРНО при правильной маске (например, 255.255.224.0).