

Дальний Восток. Версия от 19.06, 7:15

Содержание

Задача №5	3
Задача 5.1 (Дальний восток)	3
Задача №6	4
Задача 6.1 (Дальний восток)	4
Задача №7	6
Задача 7.1 (Дальний восток)	6
Задача 7.2 (Дальний восток)	6
Задача №8	8
Задача 8.1 (Дальний восток)	8
Задача 8.2 (Дальний восток)	9
Задача №11	12
Задача 11.1 (Дальний восток)	12
Задача 11.2 (Дальний восток)	13
Задача №13	15
Задача 13.1 (Дальний восток)	15
Задача 13.2 (Дальний восток)	16
Задача 13.3 (Дальний восток)	17
Задача 13.4 (Дальний восток)	18
Задача №14	20
Задача 14.1 (Дальний восток)	20
Задача №15	22
Задача 15.1 (Дальний восток)	22
Задача 15.2 (Дальний восток)	22
Задача №16	24
Задача 16.1 (Дальний восток)	24
Задача №17	25
Задача 17.1 (Дальний восток)	25
Задача №19-21	26
Задача 19.1 (Дальний восток)	26
Задача 20.1 (Дальний восток)	27

Задача 21.1 (Дальний восток)	28
Задача №23	30
Задача 23.1 (Дальний восток)	30
Задача №25	31
Задача 25.1 (Дальний восток)	31
Задача 25.2 (Дальний восток)	33
Задача 25.3 (Дальний восток)	34
Задача 25.4 (Дальний восток)	35
Задача №26	37
Задача 26.1 (Дальний восток)	37
Задача №27	38
Задача 27.1 (Дальний восток)	38

Задача №5

Задача 5.1 (Дальний восток)

Автомат получает на вход число N и преобразует его в число R .

1. Строится двоичная запись числа N .
2. Если сумма цифр двоичного числа нечетное число, то дописывается 01.
3. Если сумма цифр двоичного числа четное число, то дописывается 10.
4. Результат переводится в десятичную систему и он же является числом R .

Какое наименьшее число R , большее 123, может, быть результатом работы этой программы?

Решение

Для каждого числа построим его двоичную запись с помощью функции `bin()`, отбросив служебный префикс `'0b'` через срез `[2:]`. Сумма цифр в двоичном числе равна количеству единиц в его записи, поэтому для её подсчета воспользуемся методом `.count('1')`. Если это количество нечетно, допишем в конец строки `'01'`, а если четно – `'10'`. Полученный результат переведем обратно в десятичную систему с помощью `int(s, 2)`.

Все полученные значения R , которые превосходят 123, сохраним в список, чтобы в конце работы программы вывести из них минимальное.

```
results = []
for n in range(1000):
    s = bin(n)[2:] # Строим двоичную запись числа N
    sum_digits = s.count('1') # Считаем сумму цифр (количество единиц)

    # Дописываем окончание по правилам алгоритма
    if sum_digits % 2 != 0:
        s += '01'
    else:
        s += '10'

    r = int(s, 2) # Получаем результат R в десятичной системе

    # Если результат больше 123, сохраняем его
    if r > 123:
        results.append(r)

# Находим и выводим наименьший результат R
print(min(results))
```

Ответ: 125

Задача №6

Задача 6.1 (Дальний восток)

Исполнитель Черепаха передвигается по плоскости и оставляет след в виде линии. У исполнителя существует 6 команд: Поднять хвост, означающая переход к перемещению без рисования; Опустить хвост, означающая переход в режим рисования; Вперёд n (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; Назад n (где n – целое число), вызывающая передвижение в противоположном голове направлении; Направо m (где m – целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, Налево m (где m – целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

В начальный момент Черепаха находится в начале координат и направлена вверх (вдоль положительного направления оси ординат).

Запись **Повтори k [Команда1 Команда2 ... КомандаS]** означает, что заданная последовательность из S команд повторится k раз.

Черепахе был дан для исполнения следующий алгоритм:

Повтори 4 [Вперёд 28 Направо 90 Вперёд 26 Направо 90]

Поднять хвост

Вперёд 12 Направо 90 Вперёд 13 Налево 90

Опустить хвост

Повтори 4 [Вперёд 67 Направо 90 Вперёд 76 Направо 90]

Определите площадь пересечения фигур, ограниченного заданными алгоритмом линиями.

Решение

Напишем программу, используя библиотеку turtle для визуализации движения Черепахи по заданному алгоритму. Основная цель — построить фигуры по алгоритму, а затем проверить, какие точки с целочисленными координатами лежат внутри их пересечения. Для этого сначала выполняется алгоритм Черепахи, который рисует границы двух фигур. После этого необходимо отрисовать точки с целочисленными координатами, чтобы после завершения построения подсчитать их количество. Для удобства визуализации применяется масштабирование через переменную m , иначе фигуры будут слишком маленькими. Все перемещения вперёд умножаются на m .

```
from turtle import *

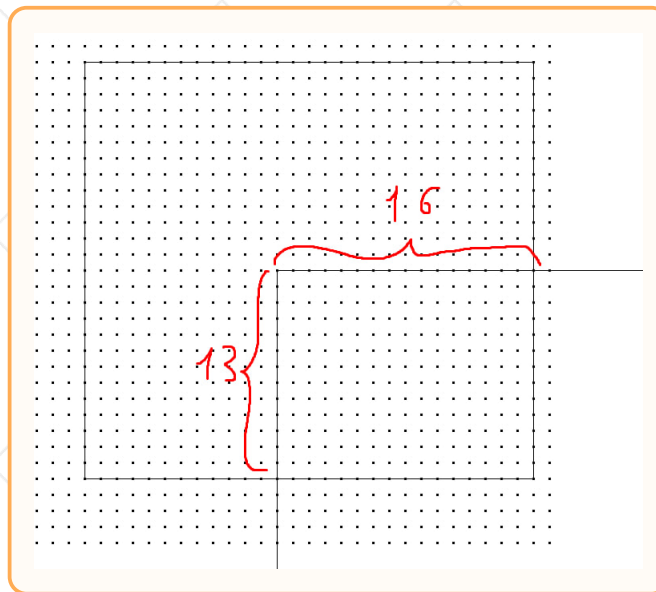
tracer(0) # Ускоряем анимацию
k = 25 # Масштаб
# Добавляем ползунки справа и снизу
screensize(3000, 3000)

for i in range(4):
    forward(28 * k) # Вперёд с учётом масштаба
    right(90)
    forward(26 * k)
```

```
right(90)
pu() # Поднимаем хвост
forward(12 * k)
right(90)
forward(13 * k)
left(90)
pd() # Опускаем хвост
for i in range(4):
    forward(67 * k)
    right(90)
    forward(76 * k)
    right(90)

pu() # Поднимаем хвост
# Отрисовываем координатную сетку
for x in range(-20, 30):
    for y in range(-30, 20):
        goto(x * k, y * k)
        dot(4) # Ставим точку
done() # Не даём программе завершиться
```

После запуска программы получаем картинку:



Осталось посчитать площадь пересечения фигур, включая точки на границах этого пересечения. Для этого нужно перемножить количество точек, вычтя 1, на сторонах нашей фигуры.

Ответ: 208

Задача №7

Задача 7.1 (Дальний восток)

Рома записывает голосовое сообщение для своего брата. Перед отправкой сообщение оцифровывается в формате моно с частотой дискретизации 48 000 Гц и глубиной кодирования 16 бит. Определите наименьшее целое количество Кбайт, необходимое для сохранения сообщения в памяти (без учёта заголовка), если его длительность - 1 минута 15 секунд. В ответе укажите только число.

Решение

Объём звука можно найти по формуле: количество каналов · частота (в Гц) · длительность записи (в секундах) · глубина кодирования. Также важно помнить типы форматов, моно содержит 1 канал. Получаем $1 \cdot 48000 \cdot 75 \cdot 16 \text{ бит} = 57\,600\,000 \text{ бит}$.

Ответ просится записать в Кбайтах, переводим, поделив на 8 и 1024: $\frac{57\,600\,000}{8 \cdot 1024} = 7031,25$ Кбайт. Округляем вверх, чтобы весь объём записи влез.

Ответ: 7032

Задача 7.2 (Дальний восток)

Настя записывала голосовое сообщение Дане. Перед отправкой сообщение оцифровывается в формате стерео с частотой дискретизации 28 000 Гц и глубиной кодирования 8 бит.

Определите наименьшее целое количество Кбайт, необходимое для сохранения сообщения в памяти (без учёта заголовка), если его длительность – 2 минуты 20 секунд. В ответе укажите только число.

Решение

Для нахождения объёма аудиофайла используется формула:

Объём = Количество каналов · Частота дискретизации · Глубина кодирования · Время

Определим параметры записи по условию:

Формат стерео означает, что запись двухканальная (количество каналов = 2).

Частота дискретизации = 28000 Гц.

Глубина кодирования = 8 бит (это ровно 1 байт).

Длительность записи = 2 минуты 20 секунд. Переведем время в секунды: $2 \cdot 60 + 20 = 140$ секунд.

Найдем объём аудиозаписи в байтах:

Объём = $2 \cdot 28000 \cdot 1 \text{ байт} \cdot 140 = 56000 \cdot 140 = 7\,840\,000 \text{ байт}$.

Переведем полученный объём в килобайты, разделив значение на 1024:

$$\frac{7840000}{1024} = 7656.25 \text{ Кбайт}$$

По условию требуется найти наименьшее целое количество Кбайт, необходимое для сохранения файла в памяти. Если округлить результат в меньшую сторону до 7656 Кбайт, то оставшаяся часть файла (0.25 Кбайт) не поместится и сообщение сохранится не полностью. Для сохранения всего файла округляем значение в большую сторону до ближайшего целого: 7656.25 Кбайт округляем до 7657 Кбайт.

Ответ: 7657

Задача №8

Задача 8.1 (Дальний восток)

Все пятибуквенные слова, составленные из букв Л, Е, Б, Е, Д, Ъ, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

БББББ

ББББД

ББББЕ

ББББЛ

ББББЬ

Определите, под каким номером в этом списке стоит последнее слово, которое начинается с гласной буквы, содержит ровно две буквы Л и не содержит ни одной буквы Ъ.

Решение

Решение руками

Для начала дадим каждой букве из алфавита числовое обозначение

Б - 0

Д - 1

Е - 2

Л - 3

Ь - 4

Мы получили числа 5сс. Теперь составим максимальное по значению число, удовлетворяющее всем условиям:

ЕЛЛЕЕ – 23322₅

Переведем данное число в 10сс и получим число 1712₁₀

Для того, чтобы узнать порядковый номер данного слова, увеличим число на 1 и получим ответ 1713.

Решение программой с помощью циклов

Будем последовательно перебирать все возможные пятибуквенные слова из заданного алфавита в том порядке, в котором они расположены в условии. Для каждого полученного слова будем увеличивать счётчик, который хранит его номер в общем списке. Затем проверим выполнение всех условий задачи: первая буква должна быть Е, слово должно содержать ровно две буквы Л и не должно содержать букву Ъ. Если все условия выполняются, выводим номер слова.

```
# Строка с буквами в алфавитном порядке
```

```
a = "БДЕЛЬ"
```

```
# Счётчик номеров слов
```

```
count = 0
```

```
# Перебираем первую букву слова
```

```
for x1 in a:
```



```
# Перебираем вторую букву слова
for x2 in a:
    # Перебираем третью букву слова
    for x3 in a:
        # Перебираем четвёртую букву слова
        for x4 in a:
            # Перебираем пятую букву слова
            for x5 in a:
                # Собираем слово из выбранных букв
                s = x1+x2+x3+x4+x5
                # Переходим к следующему номеру слова
                count += 1
                # Проверяем выполнение всех условий задачи
                if s[0] == "Е" and s.count("Л") == 2 and "Ъ" not in s:
                    print(count)
```

Вместо пяти вложенных циклов воспользуемся функцией `product()`, которая автоматически генерирует все возможные пятибуквенные слова из заданного набора букв. Поскольку буквы указаны в алфавитном порядке, функция будет выдавать слова в том же порядке, в котором они должны находиться в общем списке. Для каждого слова будем вести подсчёт номера и проверять выполнение условий задачи.

Если все условия выполняются одновременно, выводим номер слова.

```
# Импортируем функцию product для генерации всех комбинаций
from itertools import product
# Счётчик номеров слов
count = 0
# Генерируем все возможные пятибуквенные слова
for i in product("БДЕЛЬ", repeat = 5):
    # Преобразуем кортеж символов в строку
    s = "".join(i)
    # Увеличиваем номер текущего слова
    count += 1
    # Проверяем выполнение всех условий задачи
    if s[0] == "Е" and s.count("Л") == 2 and "Ъ" not in s:
        print(count)
```

Ответ: 1713

Задача 8.2 (Дальний восток)

Все шестибуквенные слова, составленные из букв К, О, М, Б, И записаны в алфавитном порядке и пронумерованы.

Ниже приведено начало списка:

1. ББББББ

2. ББББИ
3. ББББК
4. ББББМ
5. ББББО
6. БББИБ

...

Под каким номером в списке стоит последнее слово с нечётным номером, которое не начинается с буквы М, содержит не менее двух букв К и не содержит букв И?

Решение

Решение руками

Для начала дадим каждой букве из алфавита числовое обозначение

Б - 0
И - 1
К - 2
М - 3
О - 4

Мы получили числа 5сс. Теперь составим максимальное по значению число, удовлетворяющее всем условиям:

ООООКК - 444422₅

Переведем данное число в 10сс и получим число 15612₁₀

Для того чтобы узнать порядковый номер данного слова увеличим число на 1 и получим ответ 15613

Решение программой с помощью циклов

Будем последовательно перебирать все возможные шестибуквенные слова из заданного алфавита в том порядке, в котором они должны располагаться в списке. Для каждого сформированного слова будем вести подсчёт его номера. После получения очередного слова проверим выполнение всех условий задачи: слово не должно начинаться с буквы М, должно содержать не менее двух букв К, не должно содержать букву И, а его номер должен быть нечётным.

Если все условия одновременно выполняются, выводим номер слова.

```
# Записываем все допустимые буквы в алфавитном порядке
a = "БИКМО"
# Счётчик номеров слов
count = 0
# Перебираем первую букву слова
for x1 in a:
    # Перебираем вторую букву слова
    for x2 in a:
        # Перебираем третью букву слова
        for x3 in a:
```

```
# Перебираем четвёртую букву слова
for x4 in a:
    # Перебираем пятую букву слова
    for x5 in a:
        # Перебираем шестую букву слова
        for x6 in a:
            # Собираем слово из выбранных букв
            s = x1+x2+x3+x4+x5+x6
            # Увеличиваем номер текущего слова
            count += 1
            # Проверяем выполнение всех условий задачи
            if s[0] != "М" and s.count("К") >= 2 and "И" not in s and count % 2 != 0:
                print(count)
```

Решение программой с помощью модуля itertools

Вместо шести вложенных циклов воспользуемся функцией `product()` из модуля `itertools`. Эта функция автоматически генерирует все возможные слова заданной длины из указанного набора букв. Поскольку буквы в строке записаны в нужном порядке, генерация также происходит в правильном порядке нумерации слов. Для каждого слова будем вести подсчёт его номера и проверять выполнение условий задачи.

Если все условия одновременно выполняются, выводим номер слова.

```
# Импортируем функцию для генерации всех возможных комбинаций
from itertools import product
# Счётчик номеров слов
count = 0
# Генерируем все возможные шестибуквенные слова
for i in product("БИКМО", repeat = 6):
    # Преобразуем кортеж символов в строку
    s = "".join(i)
    # Увеличиваем номер текущего слова
    count += 1
    # Проверяем выполнение всех условий задачи
    if s[0] != "М" and s.count("К") >= 2 and "И" not in s and count % 2 != 0:
        print(count)
```

Ответ: 15613

Задача №11

Задача 11.1 (Дальний восток)

На предприятии каждой изготовленной детали присваивают серийный номер, состоящий из 200 символов. В базе данных каждый серийный номер занимает одинаковое и минимально возможное число байт. При этом используется посимвольное кодирование серийных номеров, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 85 536 серийных номеров потребовалось не менее 9 Мбайт памяти. Определите минимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только целое число.

Решение

Сначала переведем объем памяти в 9 Мбайт в байты:

$$9 \text{ Мбайт} = 9 \cdot 1024 \cdot 1024 = 9\,437\,184 \text{ байт}$$

По условию задачи, для хранения 85 536 серийных номеров потребовалось не менее 9 Мбайт памяти. Найдем минимальный объем памяти в байтах, который должен занимать один серийный номер (обозначим его за V):

$$\begin{aligned} 85\,536 \cdot V &\geq 9\,437\,184 \\ V &\geq \frac{9\,437\,184}{85\,536} \approx 110.33 \text{ байт.} \end{aligned}$$

Поскольку для хранения каждого серийного номера отводится целое число байт, минимально возможное значение V равно 111 байт.

Каждый серийный номер состоит из 200 символов, при этом каждый символ кодируется одинаковым и минимально возможным целым числом бит i .

Объем памяти для хранения одного серийного номера в битах равен $200 \cdot i$ бит.

Выразим этот объем в байтах:

$$V = \frac{200 \cdot i}{8} = 25 \cdot i \text{ байт.}$$

Так как число байт V кратно 25 и должно удовлетворять условию $V \geq 111$, то минимально возможным значением объема одного серийного номера является ближайшее большее или равное число, кратное 25:

$$V = 125 \text{ байт (при } i = 5 \text{ бит).}$$

При $i = 4$ бит объем одного номера составил бы $25 \cdot 4 = 100$ байт, тогда для всех номеров потребовалось бы 8 553 600 байт, что меньше 9 Мбайт.

Следовательно, минимальное число бит на символ i равно 5.

Определим минимально возможную мощность алфавита M , для посимвольного кодирования которого требуется 5 бит:

$$2^4 < M \leq 2^5, \text{ то есть } 16 < M \leq 32.$$

Минимально возможная мощность алфавита M равна 17 (так как при $M = 16$ было бы достаточно 4 бит).

Ответ: 17

Задача 11.2 (Дальний восток)

На заводе каждой изготовленной детали присваивают уникальный код, состоящий из 450 символов. В базе данных каждый серийный номер занимает одинаковое и минимально возможное число байт. При этом используется посимвольное кодирование кодов, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 780 100 таких кодов потребовалось не более 83 Мбайт памяти. Определите максимально возможную мощность алфавита, используемого для записи кодов. В ответе запишите только целое число.

Решение

Сначала переведем объём памяти в 83 Мбайта в байты: $83 \text{ Мбайт} = 83 \cdot 1024 \cdot 1024 = 87031808$ байт.

По условию задачи, для хранения 780100 серийных номеров потребовалось не более 83 Мбайт памяти. Найдем максимальный объем памяти в байтах, который должен занимать один серийный номер (обозначим его за V):

$$780100 \cdot V \leq 87031808$$

$$V \leq \frac{87031808}{780100} \approx 111.56 \text{ байт.}$$

Поскольку для хранения каждого серийного номера отводится целое число байт, максимально возможное значение V равно 111 байт.

Каждый серийный номер состоит из 450 символов, при этом каждый символ кодируется одинаковым и минимально возможным целым числом бит i . Объем памяти для хранения одного серийного номера в битах равен $450 \cdot i$ бит.

Найдём, сколько бит выходит на 1 символ:

$$i = \frac{111 \cdot 8}{450} \approx 1.97 \text{ бит.}$$

Взять $i = 2$ не получится, так как в этом случае мы получим как минимум $\frac{450 \cdot 2}{8} = 112.5$ байт > 111 байт.

Значит, нам подходит $i = 1$. Тогда максимальная мощность алфавита вычисляется по формуле:
 $N = 2^i = 2^1 = 2$.

Ответ: 2

Задача №13

Задача 13.1 (Дальний восток)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и его маске. Широковещательным адресом называется специализированный адрес, в котором на месте нулей в маске стоят единицы.

Адрес сети и широковещательный адрес не могут быть использованы для адресации сетевых устройств.

Сеть задана одним из входящих в неё узлов с IP-адресом 152.14.220.61 и маской сети 255.255.254.0. Определите широковещательный адрес и в ответе запишите сумму значений его октетов IP-адреса.

Например, если бы найденный адрес был равен 100.20.3.4, то в ответе следовало бы записать: 127.

Решение

Решение руками

Переведём IP-адрес узла и маску сети в двоичную систему счисления:

IP узла	10011000.00001110.11011100.00111101
Маска	11111111.11111111.11111110.00000000

Узлы в сети имеют вид: 10011000.00001110.1101110x.xxxxxxxx

Нам нужно найти широковещательный IP-адрес – то есть поместить на все места символа x единички:

$$10011000.00001110.11011101.11111111_2 = 152.14.221.255_{10}$$

Сумма октетов равна:

$$152 + 14 + 221 + 255 = 642$$

Решение программой

Для нахождения широковещательного IP-адреса необходимо определить адрес сети и взять последний возможный адрес.

В Python для работы с IP-адресами удобно использовать модуль `ipaddress`, который позволяет легко выполнять операции с сетевыми диапазонами. Создаем объект сети с помощью метода `ip_network()`, используя IP-адрес узла и маску подсети. Затем, используя индексацию `[-1]`, получаем последний адрес в сети, который и будет искомым широковещательным.

```
# Импортируем модуль для работы с IP-адресами
from ipaddress import *
```

```
# Создаем объект сети с указанным IP-адресом и маской
net = ip_network("152.14.220.61/255.255.254.0", 0)
```

```
# Получаем последний адрес в сети:  
# net[-1] - последний адрес (широковещательный)  
print(net[-1])  
  
# Вывод: 152.14.221.255  
# Сумма октетов: 152 + 14 + 221 + 255 = 642
```

Ответ: 642

Задача 13.2 (Дальний восток)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и его маске. Широковещательным адресом называется специализированный адрес, в котором на месте нулей в маске стоят единицы.

Адрес сети и широковещательный адрес не могут быть использованы для адресации сетевых устройств.

Сеть задана IP-адресом одного из входящих в неё узлов 167.66.136.176 и сетевой маской 255.254.0.0.

Определите наименьший IP-адрес данной сети, который может быть присвоен компьютеру. В ответе укажите сумму октетов найденного IP-адреса

Решение

Решение руками:

Сначала найдем адрес самой сети, применив поразрядную конъюнкцию (логическое «И») к IP-адресу узла и маске.

Первый байт: $167 \& 255 = 167$.

Для второго байта выполним операцию «И» в двоичном виде:

66 в двоичной системе: 01000010

254 в двоичной системе: 11111110

Перемножим разряды:

$01000010 \& 11111110 = 01000010$

Полученное число в десятичной системе равно 66.

Третий байт: $136 \& 0 = 0$.

Четвертый байт: $176 \& 0 = 0$.

Таким образом, адрес сети равен 167.66.0.0.

По условию требуется найти наименьший IP-адрес, который может быть присвоен компьютеру. Сам адрес сети (167.66.0.0) под адресацию устройств использоваться не может. Следовательно, наименьшим допустимым адресом является следующий за адресом сети: 167.66.0.1.

Сложим значения октетов найденного IP-адреса:

$$167 + 66 + 0 + 1 = 234$$

Решение Python:

Параметр `strict=0` позволяет использовать адрес узла для инициализации сети. Объект сети хранит все адреса последовательно. Под индексом 0 в ней находится адрес самой сети (167.66.0.0), а под индексом 1 — первый применимый для компьютера адрес (167.66.0.1). Преобразуем этот адрес в строку, делим по точкам на части, переводим их в целые числа и складываем.

```
from ipaddress import ip_network

# Создаем объект сети
net = ip_network('167.66.136.176/255.254.0.0', strict=0)

# Получаем первый допустимый хост в сети (индекс 1) в виде строки
min_ip = str(net[1])

# Разбиваем строку по точкам, переводим части в числа и суммируем их
ans = sum(int(x) for x in min_ip.split('.'))
print(ans)
```

Ответ: 234

Задача 13.3 (Дальний восток)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и его маске. Широковещательным адресом называется специализированный адрес, в котором на месте нулей в маске стоят единицы.

Адрес сети и широковещательный адрес не могут быть использованы для адресации сетевых устройств.

Сеть задана IP-адресом одного из входящих в неё узлов 134.128.220.14 и сетевой маской 255.192.0.0.

Определите наибольший IP-адрес данной сети, который может быть присвоен компьютеру. В ответе укажите найденный IP-адрес без разделителей.

Например, если бы найденный адрес был равен 111.22.3.44, то в ответе следовало бы записать 11122344.

Решение**Решение руками**

Переведём IP-адрес узла и маску сети в двоичную систему счисления:

IP узла	10000110.10000000.11011100.00001110
Маска	11111111.11000000.00000000.00000000

Узлы в сети имеют вид: 10000110.10xxxxxx.xxxxxxxx.xxxxxxxx

Нам нужно найти наибольший IP-адрес данной сети, который может быть присвоен компьютеру. Но мы не можем взять последний IP-адрес, который является широковещательным. Поэтому ответом будет являться:

$$10000110.10111111.11111111.11111110_2 = 134.191.255.254_{10}$$

Решение программой

Для нахождения максимального IP-адреса, который может быть назначен компьютеру, необходимо определить адрес сети и взять предпоследний возможный адрес.

В Python для работы с IP-адресами удобно использовать модуль `ipaddress`, который позволяет легко выполнять операции с сетевыми диапазонами. Создаем объект сети с помощью метода `ip_network()`, используя IP-адрес узла и маску подсети. Затем, используя индексацию `[-2]`, получаем предпоследний адрес в сети, который и будет максимальным искомым.

```
# Импортируем модуль для работы с IP-адресами
from ipaddress import *

# Создаем объект сети с указанным IP-адресом и маской
net = ip_network("134.128.220.14/255.192.0.0", 0)

# Получаем последний адрес в сети:
# net[-1] - последний адрес (широковещательный)
# net[-2] - предпоследний адрес (максимальный), который
# может быть назначен компьютеру
print(net[-2])

# Вывод: 134.191.255.254
```

Ответ: 134191255254

Задача 13.4 (Дальний восток)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и его маске. Широковещательным адресом называется специализированный адрес, в котором на месте нулей в маске стоят единицы.

Адрес сети и широковещательный адрес не могут быть использованы для адресации сетевых устройств.

Сеть задана одним из входящих в неё узлов с IP-адресом 146.180.173.153 и маской сети 255.192.0.0. Найдите наибольший в данной сети IP-адрес. В ответе укажите найденный IP-адрес без разделителей.

Например, если бы найденный адрес был равен 111.22.3.44, то в ответе следовало бы записать: 11122344.

Решение

Решение руками:

Наибольшим возможным IP-адресом в сети является её широковещательный адрес. По определению, в широковещательном адресе все биты, которые в маске равны нулю, заменяются на единицы.

Распишем IP-адрес узла 146.180.173.153 и маску 255.192.0.0 по байтам:

Первый байт маски равен 255 (все единицы), поэтому первый байт искомого адреса совпадает с первым байтом узла: 146.

Для второго байта переведем числа в двоичную систему:

180 в двоичной системе: 10110100

192 в двоичной системе: 11000000

Первые два разряда маски равны 1 (это сетевая часть), остальные шесть разрядов равны 0 (это часть номера).

Сетевая часть второго байта узла: 10.

Заменяем оставшиеся шесть хостовых нулей на единицы:

Получаем число 10111111, которое в десятичной системе равно 191.

Третий и четвертый байты маски равны 0. Заменяем все их биты на единицы:

В третьем байте получим 255.

В четвертом байте получим 255.

Таким образом, наибольший IP-адрес в данной сети равен 146.191.255.255.

Запишем его без разделителей (точек): 146191255255.

Решение Python:

Наибольший IP-адрес сети (широковещательный адрес) хранится в свойстве `broadcast_address` объекта сети. Полученный адрес преобразуем в строку и удалим из неё точки с помощью метода `replace('.', '')`.

```
from ipaddress import ip_network

# Создаем объект сети
net = ip_network('146.180.173.153/255.192.0.0', strict=0)

# Получаем широковещательный адрес (наибольший IP-адрес) в виде строки
broadcast_addr = str(net.broadcast_address)

# Удаляем точки-разделители и выводим результат
ans = broadcast_addr.replace('.', '')
print(ans)
```

Ответ: 146191255255

Задача №14

Задача 14.1 (Дальний восток)

Значение арифметического выражения

$$5^{127} + 5^{27} - x,$$

где x – натуральное число, не превышающее 3000, записали в четверичной системе счисления.

Определите наименьшее значение x , при котором в четверичной записи числа, являющегося значением данного арифметического выражения, содержится наибольшее количество нулей.

В ответе запишите число в десятичной системе счисления.

Решение

В данной программе требуется найти такое значение переменной x из диапазона от 1 до 3000, при котором в четверичной записи числа

$$5^{127} + 5^{27} - x$$

содержится максимальное количество цифр 0. Для этого будем последовательно перебирать все возможные значения x , вычислять соответствующее значение выражения, переводить полученное число в систему счисления по основанию 4 и подсчитывать количество нулей в его записи.

```
# Функция перевода числа из десятичной системы счисления в четверичную
def cc4(x):
    # Строка для накопления результата перевода
    s = ""
    # Выполняем перевод, пока число больше нуля
    while x > 0:
        # Добавляем очередную цифру четверичной записи в начало строки
        s = str(x % 4) + s
        # Переходим к следующему шагу алгоритма деления
        x = x // 4
    # Возвращаем полученную четверичную запись
    return s
# Максимальное количество найденных нулей
mx_count = 0
# Значение x, соответствующее текущему максимуму
mx = 0
# Перебираем все возможные значения x от 1 до 3000
for x in range(1,3001):
    # Вычисляем выражение и переводим результат в четверичную систему счисления
    res = cc4(5**127 + 5**27 - x)
    # Если количество нулей больше ранее найденного максимума
```

```
if res.count("0") > mx_count:
    # Запоминаем новое максимальное количество нулей
    mx_count = res.count("0")
    # Сохраняем соответствующее значение x
    mx = x
# Выводим найденное значение x
print(mx)
```

Ответ: 2666

Задача №15**Задача 15.1 (Дальний восток)**

Заданы два отрезка $P = [3; 18]$ и $Q = [11; 24]$, лежащие на числовой прямой. Укажите наименьшую возможную длину такого отрезка A , для которого логическое выражение

$$((x \in P) \wedge (x \in Q)) \rightarrow (x \in A)$$

истинно (т. е. принимает значение 1) при любом значении переменной x .

Решение

Преобразуем исходное выражение, раскрыв импликацию и применив закон де Моргана:

$$(x \notin P) \vee (x \notin Q) \vee (x \in A)$$

Приравняем известную часть к 0 и получим:

$$(x \notin P) \vee (x \notin Q) = 0$$

$$(x \in P) \wedge (x \in Q) = 1$$

Значит, выражение даёт ложь в известной части при x , принадлежащих одновременно обоим отрезкам. Нам нужно, чтобы все значения x из данного пересечения принадлежали отрезку A , поэтому необходимо взять $A = [11; 18]$.

Тогда минимальная возможная длина данного отрезка равна:

$$18 - 11 = 7$$

Ответ: 7

Задача 15.2 (Дальний восток)

На числовой прямой даны три отрезка: $P = [55; 57]$, $O = [22; 98]$ и $D = [20; 150]$. Укажите наименьшую возможную длину такого отрезка A , что логическое выражение

$$(x \notin A) \rightarrow ((x \in P) \vee (x \notin O) \vee (x \notin D))$$

истинно (т.е. принимает значение 1) при любом значении переменной x .

Решение

Преобразуем исходное выражение, раскрыв импликацию:

$$(x \in A) \vee ((x \in P) \vee (x \notin O) \vee (x \notin D))$$

Приравняем известную часть к 0 и получим:

$$(x \in P) \vee (x \notin O) \vee (x \notin D) = 0$$

$$(x \notin P) \wedge (x \in O) \wedge (x \in D) = 1$$

Значит, выражение даёт ложь в известной части при x , принадлежащих одновременно отрезкам O и D , но не принадлежащих отрезку P . Таким значениям на числовой прямой соответствуют промежутки $[22; 55)$ и $(57; 98]$. Нам нужно, чтобы все эти значения принадлежали отрезку A , поэтому необходимо взять $A = [22; 98]$. Тогда минимальная возможная длина данного отрезка равна $98 - 22 = 76$.

Ответ: 76

Задача №16**Задача 16.1 (Дальний восток)**

Алгоритм вычисления значения функции $F(n)$, где n — целое неотрицательное число, задан следующими соотношениями:

$$F(n) = 1, \text{ при } n = 1;$$

$$F(n) = (n - 1) \cdot F(n - 1), \text{ если } n > 1.$$

Определите значение выражения:

$$(F(10938)/2 - F(10937))/F(10936)$$

Решение

Решение руками:

$$\begin{aligned} \frac{F(10938) - F(10937)}{2 \cdot F(10936)} &= \frac{10937 \cdot F(10937) - F(10937)}{2 \cdot F(10936)} = \frac{10935 \cdot F(10937)}{2 \cdot F(10936)} = \\ &= \frac{10935 \cdot 10936 \cdot F(10936)}{2 \cdot F(10936)} = 59792580 \end{aligned}$$

Ответ: 59792580

Задача №17

Задача 17.1 (Дальний восток)

В файле содержится последовательность целых чисел. Её элементы могут принимать целые значения от -100000 до 100000 включительно. Определите количество троек последовательности, в которых сумма элементов делится на максимальный элемент последовательности, оканчивающийся на 3. В ответе запишите количество найденных троек, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Решение

```
# Открываем файл '17.txt' для чтения
f = open('17.txt')
# Читаем все строки из файла, преобразуем каждую в целое число
a = [int(x) for x in f]
# Находим максимальное число среди всех элементов списка a,
# у которых последняя цифра (модуль числа) равна 3
m3 = max(x for x in a if abs(x)%10==3)
maxi = -10**10
c = 0
# Перебираем все тройки подряд идущих чисел в списке a
for i in range(len(a)-2):
    # Берём тройку чисел: a[i], a[i+1], a[i+2]
    t = a[i:i+3]
    # Проверяем, делится ли сумма элементов тройки нацело на m3
    if sum(t) % m3 == 0:
        # Если да, то увеличиваем счётчик на 1
        c += 1
        # Обновляем максимальную сумму тройки,
        # если текущая сумма больше сохранённой
        maxi = max(maxi, sum(t))

print(c, maxi) # Выводим полученные результаты на экран
```

Задача №19-21

Задача 19.1 (Дальний восток)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может:

- добавить в одну из куч (по своему выбору) 3 камня;
- увеличить количество камней в одной из куч (по своему выбору) в 2 раза.

Например, пусть в одной куче 20 камней, а в другой 30 камней; такую позицию в игре обозначим $(20, 30)$. Тогда за один ход можно получить любую из четырёх позиций: $(23, 30)$, $(20, 33)$, $(40, 30)$, $(20, 60)$.

Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в двух кучах становится не менее 173. Победителем считается игрок, сделавший последний ход, то есть первым получивший такую игровую позицию, при которой в двух кучах суммарно 173 камня или больше. В начальный момент в первой куче 27 камней, во второй куче – S камней; $1 \leq S \leq 150$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Известно, что Ваня выиграл своим первым ходом после неудачного хода Пети. Укажите минимальное значение S , при котором это возможно.

Решение

Решение руками

Чтобы Ваня победил своим первым ходом, Петя должен совершить такой неудачный ход, после которого Ваня следующим же действием сможет набрать в сумме не менее 173 камней. Быстрее всего увеличить кучу можно с помощью удваивания. Сначала Петя удваивает вторую кучу S , а затем Ваня удваивает полученную кучу, в то время как первая куча остается без изменений (27 камней). При каких S будет выполняться неравенство $2 \cdot 2 \cdot S + 27 \geq 173$?

$$4 \cdot S + 27 \geq 173$$

$$4 \cdot S \geq 146$$

$$S \geq 36.5$$

Минимальное целое значение S , удовлетворяющее этому условию, равно 37.

Решение программой

Для решения на Python используем рекурсию с проверкой всех возможных ходов $(+3, \cdot 2)$, чтобы определить, чья это выигрышная позиция. Функция возвращает 0, если игра уже завершена (камней в сумме ≥ 173), положительное число – если при оптимальной игре побеждает текущий игрок, и отрицательное – если побеждает соперник. Так как первый ход делает Петя, цикл запускается от его лица: если значение функции положительное, выигрывает Петя, а если отрицательное – Ваня.

Если функция вернула 1, то Петя победил первым ходом. Но он походил неудачно и передал победу Ване. Рассмотрим все ходы Пети и проверим, что после хотя бы одного из них Ваня мог выиграть. Также важно учесть, что Петя не должен гарантированно проигрывать, иначе нельзя назвать его ход неудачным.

При переборе возможных значений программа выводит минимум 37.

```
from functools import lru_cache

@lru_cache(None)
def game(first_heap, second_heap):
    if first_heap + second_heap >= 173:
        return 0 # Прекращаем игру
    moves = [
        game(first_heap + 3, second_heap),
        game(first_heap, second_heap + 3),
        game(first_heap * 2, second_heap),
        game(first_heap, second_heap * 2),
    ] # Генерация всех возможных ходов
    petya_win = [i for i in moves if i <= 0]
    if petya_win:
        return -max(petya_win) + 1
    else:
        return -max(moves)

for s in range(1, 151):
    # Если в данной позиции после неудачного хода Пети возможен выигрыш Вани
    # и Петя не был в гарантированно проигрышной позиции
    if ((game(27 + 3, s) == 1 or game(27, s + 3) == 1
         or game(27 * 2, s) == 1 or game(27, s * 2) == 1)
        and game(27, s) != -1):
        print(s) # Вывод минимального значения S
        break
```

Ответ: 37

Задача 20.1 (Дальний восток)

Найдите наименьшее и наибольшее значения S , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от ходов Вани.

Решение

Для решения на Python используем рекурсию с проверкой всех возможных ходов (+3, ·2), чтобы определить, чья это выигрышная позиция. Функция возвращает 0, если игра уже завершена (камней в сумме ≥ 173), положительное число – если при оптимальной игре побеждает текущий игрок, и отрицательное – если побеждает соперник.

Так как первый ход делает Петя, цикл запускается от его лица: если значение функции положительное, выигрывает Петя, а если отрицательное – Ваня.

Если функция вернула 2, то Петя победил вторым ходом. Программа перебирает и выводит значения S , в ответ берем наименьшее и наибольшее.

```
from functools import lru_cache

@lru_cache(None)
def game(first_heap, second_heap):
    if first_heap + second_heap >= 173:
        return 0 # Прекращаем игру
    moves = [
        game(first_heap + 3, second_heap),
        game(first_heap, second_heap + 3),
        game(first_heap * 2, second_heap),
        game(first_heap, second_heap * 2),
    ] # Генерация всех возможных ходов
    petya_win = [i for i in moves if i <= 0]
    if petya_win:
        return -max(petya_win) + 1
    else:
        return -max(moves)

for s in range(1, 151):
    if game(27, s) == 2:
        print(s)
```

Ответ: 36 71

Задача 21.1 (Дальний восток)

Найдите минимальное значение S , при котором:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, гарантирующей выигрыш первым ходом.

Решение

Для решения на Python используем рекурсию с проверкой всех возможных ходов (+3, *2), чтобы определить, чья это выигрышная позиция.

Функция возвращает 0, если игра уже завершена (камней в сумме ≥ 173), положительное число – если при оптимальной игре побеждает текущий игрок, и отрицательное – если побеждает соперник.

Так как первый ход делает Петя, цикл запускается от его лица: если значение функции положительное, выигрывает Петя, а если отрицательное – Ваня.

Если функция вернула -2, то Ваня гарантированно побеждает своим первым или вторым ходом. Программа перебирает значения S и выводит минимальное подходящее число.



```
from functools import lru_cache

@lru_cache(None)
def game(first_heap, second_heap):
    if first_heap + second_heap >= 173:
        return 0 # Прекращаем игру
    moves = [
        game(first_heap + 3, second_heap),
        game(first_heap, second_heap + 3),
        game(first_heap * 2, second_heap),
        game(first_heap, second_heap * 2),
    ] # Генерация всех возможных ходов
    petya_win = [i for i in moves if i <= 0]
    if petya_win:
        return -max(petya_win) + 1
    else:
        return -max(moves)

# Поиск минимального значения S для выигрыша Вани первым или вторым ходом
for s in range(1, 151):
    if game(27, s) == -2:
        print(s) # Вывод минимального значения S
        break
```

Ответ: 67

Задача №23

Задача 23.1 (Дальний восток)

Исполнитель преобразует число на экране. У исполнителя есть две команды, которые обозначены номерами:

1. Прибавь 2
2. Замени 1 на 3

Первая из этих команд увеличивает число на экране на 2. Вторая команда может применяться только к числу, в десятичной записи которого содержится хотя бы одна цифра «1», и действует, заменяя все цифры «1» в записи числа на цифры «3» (например, число 11 превратится в 33, а 21 - в 23).

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 10 результатом является число 44?

Решение

```
def f(a, b):
    if a > b:
        return 0
    if a == b:
        return 1
    t = f(a + 2, b)
    if '1' in str(a):
        t += f(int(str(a).replace('1', '3')), b)
    return t

print(f(10, 44))
```

Ответ: 6

Задача №25

Задача 25.1 (Дальний восток)

Пусть M – сумма минимального и максимального простых натуральных делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то значение M считается равным нулю.

Напишите программу, которая перебирает целые числа, большие 5 800 000, в порядке возрастания и ищет среди них такие, для которых M больше 80 000 и является палиндромом, т.е. одинаково читается слева направо и справа налево.

В ответе запишите в первом столбце таблицы первые пять найденных чисел в порядке возрастания, а во втором столбце – соответствующие им значения M . Например, для числа 298 $M = 2 + 149 = 151$.

Решение

Будем последовательно перебирать числа из заданного диапазона и для каждого числа находить его простые делители. Из найденных простых делителей определим минимальный и максимальный, после чего вычислим их сумму. Затем проверим дополнительные условия задачи: полученная сумма должна быть больше 80000 и являться палиндромом. Все подходящие числа будем выводить сразу после нахождения. После получения первых пяти результатов перебор будет остановлен.

Функция для поиска всех делителей числа, кроме 1 и самого числа

```
def divs(x):
    # Создаём множество для хранения делителей
    d = set()
    # Перебираем возможные делители от 2 до корня числа
    for i in range(2, int(x**0.5)+1):
        # Проверяем, делится ли число без остатка
        if x % i == 0:
            # Добавляем найденный делитель
            d.add(i)
            # Добавляем парный делитель
            d.add(x//i)
    # Возвращаем отсортированный список делителей
    return sorted(d)
```

Функция проверки числа на простоту

```
def is_prime(x):
    # Перебираем возможные делители от 2 до корня числа
    for i in range(2, int(x**0.5)+1):
        # Если найден делитель, число не является простым
        if x % i == 0:
            return False
    # Возвращаем результат проверки
    return x > 1
```

```
# Функция поиска минимального и максимального простых делителей
def find_m(x):
    # Начальное значение для поиска минимального простого делителя
    mn_d = 10**10
    # Начальное значение для поиска максимального простого делителя
    mx_d = -1
    # Перебираем все делители числа
    for i in divs(x):
        # Если делитель простой, обновляем минимум
        if is_prime(i):
            mn_d = min(mn_d, i)
        # Если парный делитель простой, обновляем максимум
        if is_prime(x//i):
            mx_d = max(mx_d, x//i)
        # Если оба значения найдены, возвращаем результат
        if mn_d < 10**10 and mx_d > 0:
            return mn_d, mx_d
    # Если подходящие значения не найдены, возвращаем специальные значения
    return -1, -1

count = 0 # Счётчик найденных чисел
# Перебираем числа из заданного диапазона
for i in range(5_800_001, 10_000_000):
    # Находим минимальный и максимальный простые делители
    mn_d, mx_d = find_m(i)
    # Проверяем, что значения были найдены
    if mn_d > 0 and mx_d > 0:
        # Вычисляем сумму найденных делителей
        m = mn_d + mx_d
        # Проверяем условия задачи
        if m > 80_000 and str(m) == str(m)[::-1]:
            # Выводим найденное число и вычисленное значение
            print(i, m)
            # Увеличиваем количество найденных результатов
            count += 1
            # После нахождения пяти чисел завершаем перебор
            if count == 5:
                break
```

Ответ:

5800020 96669
5801724 161161
5804045 89298

5805494 93639
5806156 1451541

Задача 25.2 (Дальний восток)

Пусть M – сумма минимального и максимального простых натуральных делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то значение M считается равным нулю.

Напишите программу, которая перебирает целые числа, большие 7 800 000, в порядке возрастания и ищет среди них такие, для которых M больше 100 000 и является палиндромом, т.е. одинаково читается слева направо и справа налево. В ответе запишите в первом столбце таблицы первые пять найденных чисел в порядке возрастания, а во втором столбце – соответствующие им значения M .

Например, для числа 298 $M = 2 + 149 = 151$.

Решение

Сначала вводится функция $is_prime(n)$, которая проверяет число на простоту. Она перебирает возможные делители от 2 до корня из числа: если находится хотя бы один делитель, то число нам не подходит, возвращаем ложь. Далее вводится функция $find_m(n)$, которая находит значение M для текущего числа. Для этого она проверяет все возможные делители: первые найденные простые числа i и $n//i$ будут подходить, так как при продолжении цикла значения будут только уменьшаться. Если минимальный или максимальный простые делители не будут найдены, то функция вернёт -1 . В конце остаётся перебрать числа, превышающие 7800000, и выбрать из них первые 5, для которых число $M > 100000$, а также является палиндромом. Чтобы реализовать вторую проверку, используем разворот строки: если число M в виде строки совпадает со своей перевёрнутой записью, значит оно является палиндромом.

```
def is_prime(n): # Функция проверки числа на простоту
    for i in range(2, int(n ** 0.5) + 1): # Перебираем возможные делители числа n
        if n % i == 0: # Если число n поделилось на i,
            return False # возвращаем ложь - оно не простое
    return n > 1 # Возвращаем истину, если число не является 1,
                #при этом цикл выше закончился

def find_m(n): # Функция нахождения числа M
    min_d = max_d = 0
    for i in range(2, int(n ** 0.5) + 1): # Перебираем возможные делители числа n
        if n % i == 0: # Если число n поделилось на i, то проверяем основной делитель
            # и парный ему на простоту:
            if is_prime(i) and min_d == 0: # Если i - простое число,
                #при этом min_d ещё не найден,
                min_d = i # записываем его в качестве мин. простого делителя
            if is_prime(n // i) and max_d == 0: # Если n // i - простое число,
```

```

# при этом max_d ещё не найден,
max_d = n // i # записываем его в качестве макс. простого делителя
if min_d > 0 and max_d > 0: # Если искомые делители нашлись,
    return min_d + max_d # возвращаем значение m
return -1 # иначе возвращаем значение -1

с = 0 # Задаём счётчик количества выведенных строк
# Перебираем числа, превышающие 7_800_000
for x in range(7_800_000 + 1, 10_000_000_000):
    m = find_m(x) # Находим число M
    # Если m больше 80000 и является палиндромом,
    if m > 100000 and str(m) == str(m)[::-1]:
        print(x, m) # выводим искомые значения на экран
        с += 1 # увеличиваем счётчик на 1
    if с == 5: # Если количество выведенных строк равно 5
        break # останавливаем перебор

```

Ответ:

```

7806926 105501
7809177 2603062
7812174 1302031
7814182 3907093
7814209 601106

```

Задача 25.3 (Дальний восток)

Пусть S – сумма всех натуральных делителей целого числа. Если таких делителей у числа нет, то значение S считается равным нулю. Напишите программу, которая перебирает целые числа, большие 8 494 154, в порядке возрастания и ищет среди них такие, у которых есть ровно 4 различных натуральных делителя, а значение S является палиндромом (то есть читается слева-направо и справа-налево одинаково).

В ответе запишите в первом столбце таблицы первые 5 найденных чисел в порядке возрастания, а во втором столбце – соответствующие им значения S . Например, для числа 20 $S = 1 + 2 + 4 + 5 + 10 + 20 = 42$.

Решение

```

# Счётчик найденных чисел
count = 0

# Перебираем числа из заданного диапазона
for x in range(8_494_154 + 1, 10_000_000):
    d = set() # множество для хранения делителей

```



```
# Флаг для преждевременного выхода из цикла,  
# если делителей стало больше 4  
flag = False  
  
# Перебираем всевозможные делители  
for i in range(1, int(x ** 0.5) + 1):  
    # Проверяем, делится ли число без остатка  
    if x % i == 0:  
        # Добавляем найденный делитель  
        d.add(i)  
        # Добавляем парный делитель  
        d.add(x // i)  
  
        if len(d) > 4: # если делителей больше чем 4  
            flag = True  
            break # останавливаем перебор  
  
if flag:  
    continue # переходим к следующему числу  
  
if len(d) == 4: # если делителей ровно 4  
    S = sum(d)  
  
    if str(S) == str(S)[::-1]: # если S - палиндром  
        print(x, S)  
        # Увеличиваем количество найденных результатов  
        count += 1  
        # После нахождения пяти чисел завершаем перебор  
        if count == 5:  
            break
```

Ответ:

8495303 8864688
8519453 8889888
8580323 8616168
8602333 8812188
8602883 8610168

Задача 25.4 (Дальний восток)

Пусть R – сумма 4 наибольших делителей числа. Напишите программу, которая перебирает целые числа, большие 1151 996, в порядке возрастания и ищет среди них такие, для которых R является простым числом и палиндромом, т.е. одинаково читается слева направо и справа налево. В ответе запишите в первом столбце таблицы первые пять найденных чисел в порядке

возрастания, а во втором столбце – соответствующие им значения R . Количество строк в таблице для ответа избыточно.

Задача №26

Задача 26.1 (Дальний восток)

Производитель детского питания производит оптовую закупку винограда у фермерских хозяйств региона. Используется виноград двух типов А и В – светлый и темный. На закупку выделена определённая сумма денег.

У фермерских хозяйств каждая партия винограда имеет свою стоимость в рублях. На выделенные деньги необходимо приобрести как можно больше винограда типа А независимо от партии и не менее одной партии винограда В. Если у фермерских хозяйств закончится виноград А, то на оставшиеся деньги необходимо приобрести как можно больше винограда В.

Если существует несколько способов закупить максимальное количество винограда следует выбрать такой при котором будет приобретено как можно больше винограда А и при этом потрачено наименьшее количество денег .

Известны выделенная для закупки сумма, а также количество и цена различных партий винограда у фермерских хозяйств. Определите для максимального количества купленного винограда наибольшее возможное число партий винограда А, а так же оставшуюся сумму денег.

Входные данные.

Первая строка входного файла содержит два целых числа: N — общее количество партий винограда у фермерских хозяйств и M — сумма денег, выделенных на закупку (в рублях). Каждая из следующих N строк описывает одну партию и содержит целое число (стоимость партии в рублях) и один символ (латинская буква А или В), определяющий тип винограда. Все данные в строках входного файла отделены одним пробелом.

Выходные данные.

В ответе запишите два целых числа: сначала наибольшее возможное число партий винограда А, затем оставшуюся сумму денег.

Пример входного файла:

```
6 110
40 В
50 А
50 В
30 В
20 А
10 В
```

В данном случае можно купить не более четырех партий винограда, из них не более трех партий типа А. Минимальная цена такой покупки 100 рублей (покупаем партии 10 В 20 А 30 В 50 А).

Остнется 0 рублей. Ответ: 2 0.

Задача №27

Задача 27.1 (Дальний восток)

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, лежащий внутри прямоугольника высотой H и шириной W . Каждая звезда обязательно принадлежит только одному из кластеров.

Истинный центр кластера, или центроид, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна.

Под расстоянием понимается расстояние Евклида между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ на плоскости, которое вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

В файле А хранятся данные о звёздах **двух** кластеров, где $H = 6.5$, $W = 4,5$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y . Значения даны в условных единицах. Известно, что количество звёзд не превышает 1000.

В файле В хранятся данные о звёздах **трех** кластеров, где $H = 5$, $W = 4$ для каждого кластера. Известно, что количество звёзд не превышает 10000. Структура хранения информации о звездах в файле В аналогична файлу А.

Известно, что в файле имеются координаты ровно трёх «лишних» точек, представляющих аномалии, которые возникли в результате помех при передаче данных. Эти точки не относятся ни к одному из кластеров, их учитывать не нужно.

Для файла А определите координаты центра каждого кластера, затем найдите два числа: P_x – минимальную из абсцисс центров кластеров, и P_y – минимальную из ординат центров кластеров.

Для файла Б определите координаты центра каждого кластера, затем найдите два числа: Q_1 – минимальное расстояние между центрами кластеров, и Q_2 – максимальное расстояние между центрами кластеров.

В ответе запишите четыре числа: сначала абсолютную величину целой части произведения произведения $P_x \times 10000$, затем абсолютную величину целой части произведения $P_y \times 10000$, затем абсолютную величину целой части произведения $Q_1 \times 10000$, затем абсолютную величину целой части произведения $Q_2 \times 10000$.