Ты с ДВ? Присылай задачи сюда <a href="https://t.me/rodya\_inf\_support">https://t.me/rodya\_inf\_support</a>

Привет всем чатам, по которым гуляет ссылка. Люблю вас, удачи на экзамене, надеюсь у вас все получится

Стрим Информатика Роди, Алексея Кабанова и Максима Рейма проходит тут:

https://www.youtube.com/live/6fQsaemrW34?feature=share

# РАЗБОР ДАЛЬНЕГО ВОСТОКА

#### Памятка к задачам:

- Ничего нового, дефолтная задача без подвохов
- Слегка изменённое условие/подвох в вопросе/подводные камни и тд (не гробы)
- Новое условие/возможные гробы и тд
- №2 база и №1 база
  - **●**Задание №3

3 общая сумма с продаж товара определенного товара на определенной улице



#### Задание 5

На вход алгоритма подаётся натуральное число N. Алгоритм строит по нему новое число R следующим образом.

- 1. Строится троичная запись числа N.
- 2. Далее эта запись обрабатывается по следующему правилу:
  - а) если число N делится на 3, то в этой записи дописываются справа две последние цифры троичной записи;
  - 6) если число N на 3 не делится, то остаток от деления умножается на 5, переводится в троичную запись и дописывается в конец числа.

Полученная таким образом запись является троичной записью искомого числа *R*.

3. Результат переводится в десятичную систему и выводится на экран.

*Например*, для исходного числа  $12 = 110_3$  результатом является число  $11010_3 = 111$ , а для исходного числа  $4 = 11_3$  результатом является число  $1112_3 = 41$ .

Укажите минимальное число N, после обработки которого с помощью этого алгоритма получается число R, не меньшее чем 290.

В ответе запишите это число в десятичной системе счисления.

#### Решение:

#### from string import printable

```
def tr(x):
    s = ''
    while x:
    s = printable[x % 3] + s
    x //= 3
    return s

def f(N):
    x = tr(N)
    if N % 3 == 0:
        x += x[-2:]
    else:
        x += tr(N % 3 * 5)
    return int(x, 3)

print(min(N for N in range(1, 10000) if f(N) >= 290))
```

6 пересечение 2 прямоугольников, нужно посчитать количество точек внутри включая точки на линиях



**●**Задание №7

фотографии размер 1920\*1080 палитра из 2^22 цветов для экономии трафика 1280\*1024 пикселей глубина цвета 21 бит сколько кбайт экономится при передаче 120 фото

```
1 #до
2 k1 = 1920 * 1080
3 i1 = 22
4 I1 = k1 * i1 * 120
5 #ПОСЛЕ
6 k2 = 1280 * 1024
7 i2 = 21
8 I2 = k2 * i2 * 120
9
10 print((I1 - I2) / 8/ 1024)
11
12 kek ×
|:
"/Applications/Python 3.13/I
265050.0
```

Даны 8 букв, уже не помню какие, но они в алфавитном порядке и нужно найти максимальный нечётный номер строки, который не начинается на определенные буквы и содержит 1 другую букву

```
from itertools import product

n = 0

for x in product(sorted('KOM∏bЮTEP'), repeat=6):

n += 1

x = ''.join(x)

if x[0] not in 'OЮE' and x.count('K') == 1:

if n % 2 != 0:

print(n)
```

# **●**Задание №9





Задание выполняется с использованием прилагаемых файлов.

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите среднее арифметическое чисел всех строк таблицы, содержащих числа, для которых выполнены оба условия:

- в строке есть одно число, которое повторяется трижды, остальные три числа различны;
- повторяющееся число строки меньше, чем удвоенное минимальное из её неповторяющихся чисел.

В ответе запишите только целую часть полученного числа.

#### Решение:

```
1    r = []
2    for line in open('9.txt'):
3     a = [int(x) for x in line.split()]
4    p3 = [x for x in a if a.count(x) == 3]
5    np = [x for x in a if a.count(x) == 1]
6    if len(p3) == 3 and len(np) == 3:
7    if p3[0] < min(np) * 2:
8     r += a
9    print(sum(r) // len(r))</pre>
```

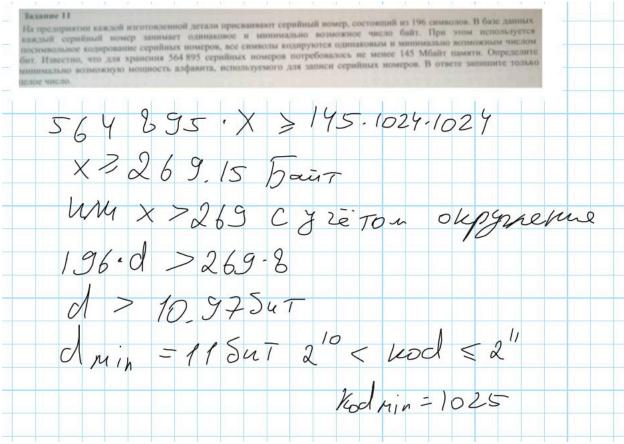
# ■Задание №10 (НЕ ЗАБЫВАЙТЕ ПРО ДЕФИСЫ) 10 нужно посчитать количество слов "сам" в составе других

# ●Задание №11

# 3410 символов одинаково минимальны число байт для хранения 2984523 потребовалось не менее 14 мегабайт найти минимальную возможную мощность алфавита для записи серийного номера

Задание 11

На предприятии каждой изготовленной детали присваивают серийный номер, состоящий из 196 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 564 895 серийных номеров потребовалось не менее 145 Мбайт памяти. Определите минимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только пелое число.



```
python
from math import *

for kod in range(1,10000):
    bit = ceil(log2(kod))
    byte = ceil(196*bit/8)
    if 564_895*byte >= 145*1024*1024:
        print(kod)
        break
```

база, строка с 7 далее n(3;10000) 5. Нужно найти минимальный n для которого сумма цифр строки будет 49

```
for n in range(4, 10000):
    s = '7' + n * '5'
    while ...:
    if sum(int(x) for x in s) == 49:
        print(n)
        break
```

# **●**Задание №13

13 дала маска и узел, нужно найти максимальный адрес, который можно дать пк

```
from ipaddress import *

net = ip_network(address: '98.81.154.195/255.252.0.0', strict: 0)

for ip in net.hosts():
    print(ip)
```

#### № 21413 Досрочная волна 2025 (Уровень: Базовый)

Операнды арифметического выражения записаны в системе счисления с основанием 21.  $82934x2_{21}+2924xx7_{21}+67564x8_{21}$ 

В записи чисел переменной х обозначена неизвестная цифра из алфавита 21-ричной системы счисления. Определите наименьшее значение х, при котором значение данного арифметического выражения кратно 20. Для найденного значения х вычислите частное от деления значения арифметического выражения на 20 и укажите его в ответе в десятичной системе счисления. Основание системы счисления указывать не нужно.

Показать ответ

```
from string import printable

for x in printable[:21]:
    x1 = int(f'82934{x}2', 21)
    x2 = int(f'2924{x}{x}7', 21)
    x3 = int(f'67564{x}8', 21)
    if (x1 + x2 + x3) % 20 == 0:
        print((x1 + x2 + x3) // 20)

print((x1 + x2 + x3) // 20)
```

# ●Задание №15

#### 15 неравенства а,х,у

```
def f(x, y, A):
    return (x <= 19) or (y < 2 * x + A - 50) or (y > 17)

print(min([A for A in range(0, 200) if all(f(x, y, A) == 1 for x in range(0, 400) for y in range(0, 400))]))

# 2000 * 2000 * 1000 = 4 000 000 000
```

#### ■Задание №16

16 база с лру кешем, ноут не взорвал

#### ●Задание №17

Один элемент будет 2 значным и сумма элементов должна быть больше максимального элемента, который оканчивается на 19

#### **—**Задание №18

18 с угловыми из левой в правую нижнюю, есть ещё один угол, который смотрит в левый верхний угол и нужно удалить там значения

# **●**3адание №19-21

#### 19-21 одна куча на убывание

```
def f(x,p):
    if x<=87: return p%2==0
    if p==0: return 0
    h=[f(x-2,p-1),f(x//2,p-1)]
    return any(h) if p%2!=0 else all(h)
print('19)', [x for x in range(89,10000) if f(x,2)])
print('20)', [x for x in range(89,10000) if not f(x,1)
and f(x,3)])
print('21)', [x for x in range(89,10000) if not f(x,2)
and f(x,4)])</pre>
```

#### —Задание №22 (№8376 на компегэ)

25 процессов и нужно найти количество процессов в 14 мс (разрисовать единички/прямоугольники, но ничего не нужно двигать) примерно 55 минута ЕГЭ

●Задание №23 база дефолт изи пизи лемон сквизи)

#### ●Задание №24

Нужно найти максимальное количество символов, где D встречается 1 раз и он вначале, и содержится 50 цифр

```
python
s = open('24.txt').readline()

l = m = k = 0

for r in range(len(s)):
    if s[r]=='D':
        l = r
        k = 0
    if s[r] in '0123456789': k+=1

    if k==50: m = max(m, r-l+1)
print(m)
```

#### **—**Задание №25

нужно найти числа, большие 2.900.000 +-, которые разбиваются только на 2 простых множителя, которые содержат в себе ровно один 0 нужно найти первые 5 чисел и их максимальный множитель



#### (https://vkvideo.ru/video-205865487\_456239730?t=12m27s)

# !!! ЕСЛИ ДВА ЦВЕТА, ТО ЭКСЕЛЬ. ЕСЛИ ТРИ - ТО КОДОМ (№5988 с компегэ три цвета)

# Это коробки с 2 данными Длина коробки и материал (0;1) нужно найти максимальное количество коробок, которые друг в друга надо засунуть и чтобы материал у соседних пар был разный

Входные данные1

В первой строке входного файла находятся число N — количество коробок красного цвета в магазине (натуральное число, не превышающее 10 000) и через пробел число M — количество коробок синего цвета в магазине (натуральное число, не превышающее 10 000). В следующих N строках находятся значения длин сторон коробок красного цвета (все числа натуральные, не превышающие 10 000) и через знак табуляции значения длин сторон коробок синего цвета (все числа натуральные, не превышающие 10 000), каждая пара таких значений — в отдельной строке; в последних N-M строках второе число опускается, и числа, соответствующие длинам сторон коробок красного цвета, идут каждое в отдельной строке.

Запишите в ответе два целых числа: сначала наибольшее количество коробок, которое можно использовать для упаковки одного подарка, затем максимально возможную длину стороны самой маленькой коробки в таком наборе.

Типовой пример организации данных во входном файле

50

Пример входного файла приведён для случая пяти коробок красного цвета и четырёх коробок синего цвета, когда минимальная допустимая разница между длинами сторон коробок, подходящих для упаковки «матрёшкой», составляет 3 единицы. При таких исходных данных условию задачи удовлетворяют наборы коробок с длинами сторон 39, 42, 50 и 55 или 40, 44, 50 и 55 соответственно, т. е. количество коробок равно 4, а длина стороны самой маленькой коробки равна 40.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемых файлов.

```
python

f = open('26.txt')
N, M = [int(x) for x in f.readline().split()]
gruz = [int(f.readline()) for i in range(N)]
kont = [int(f.readline()) for i in range(M)]

gruz.sort(reverse=1)

a = []
for x in kont:
    mg = max([y for y in gruz if y<=x],default=0)
    if mg!=0:
        a.append(mg)
        gruz.remove(mg)

sorted(a)
print(len(a), a[-1]-a[-2])</pre>
```

```
with open('26.txt') as f:
  n, m = map(int, f.readline().split())
  ws = [int(f.readline()) for _ in range(n)]
  ks = [int(f.readline()) for _ in range(m)]
ws.sort()
ks.sort()
res = []
mask = [0]*m
for w in ws:
  for i in range(m):
    if mask[i] == 0 and w \le ks[i]:
      res += w
      mask[i] = 1
      break
mx_weight = max(w for w in ws if w <= ks[-1])</pre>
print(len(res), mx_weight - res[-1])
```



#### Задание выполняется с использованием прилагаемых файлов.

26

Входной файл содержит сведения о массе грузов, поступивших в транспортную компанию, и о параметрах контейнеров, которые у неё имеются. В один контейнер может быть упакован только один груз. Определите максимальное количество грузов, которое может быть упаковано, и максимально возможную разность двух самых тяжёлых грузов в этом случае.

#### Входные данные

В первой строке входного файла находятся два натуральных числа N ( $N \leq 1000$ ) и M ( $M \leq 1000$ ) — количество грузов и количество контейнеров соответственно. Следующие N строк содержат числа, обозначающие массы грузов, затем идут M строк, где указана максимально допустимая масса груза для размещения в конкретном контейнере. Числа M и N могут быть не равны. Запишите в ответе два натуральных числа: сначала максимальное количество грузов, которое может быть упаковано, затем максимально возможную разность двух самых тяжёлых грузов в этом случае.

Типовой пример организации данных во входном файле
56
170
200
130
170
100
150
150
180
130
130
200
При таких исходных данных максимальное количеств
The state of the s

При таких исходных данных максимальное количество грузов, которое может быть упаковано в контейнеры, равно 4, если взять грузы массой 110, 130, 170 и 170 соответственно. Максимальная разность двух самых тяжёлых грузов в наборе составит 30, если взять грузы массой 110, 130, 170 и 200 соответственно.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемых файлов.

Ответ:	

27 файл А: Обычная кластеризация без всего на 2 кластера Файл Б: Будет 3 кластера и 3 аномальные точки, которые надо убрать

Все задание на центры, для файла А: абсолютное значение ар. X, v

Б: абсолютное значение суммы абсцисс и ординат центров максимально из минимального кластеров