

Краткие рекомендации по решению заданий КЕГЭ по информатике (2024)

1 Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы) – задание со схемой дорог и таблицей

2 Умение строить таблицы истинности и логические схемы

Для построения таблицы истинности лучше написать небольшую программу на Питон. Например, дана логическая функция $(x \equiv \neg y) \rightarrow ((x \wedge w) \equiv (z \wedge \neg w))$.

```
for x in range(2):
    for y in range(2):
        for w in range(2):
            for z in range(2):
                if ((x == (not y)) <= ((x and w) == (z and (not w)))) == 0:
                    print(x,y,w,z)
```

$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \equiv y$	$\neg x$
x and y	x or y	(not x) or y или $x <= y$	$x == y$	not x

Очень важно:

- переменные с *not* брать в скобки, так как у логических операторов приоритет в Питоне ниже чем у операторов сравнения и равенства, а по законам логики у *not* самый высокий приоритет
- если всё выражение истинно, можно не писать в конце «==1:»
- если всё выражение ложно, необходимо взять его в скобки и в конце обязательно написать «==0:»

3 Знание о технологии хранения, поиска и сортировки информации в реляционных базах данных – задание на поиск в базе данных. Очень часто удобно использовать функцию ВПР.

Аргументы функции

ВПР

Искомое_значение выделяем ячейку с искомым значением ↑ = любое

Таблица выделяем всю таблицу для поиска ↑ = число

Номер_столбца отсчитываем и вводим номер столбца ↑ = число

Интервальный_просмотр всегда пишем ноль ↑ = логическое

=

Ищет значение в крайнем левом столбце таблицы и возвращает значение ячейки, находящейся в указанном столбце той же строки. По умолчанию таблица должна быть отсортирована по возрастанию.

Номер_столбца номер столбца в таблице, из которого нужно вернуть значение. Первый столбец значений таблицы имеет номер 1.

Значение:

[Справка по этой функции](#)

4 Умение кодировать и декодировать информацию. Как правило, задача на условие Фано и лучше построить двоичное дерево

5 Формальное исполнение алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд.

1. Если алгоритм предполагает перевод чисел из десятичной системы в двоичную и из двоичной в десятичную, то удобно воспользоваться функциями Excel (категория «Инженерные»): ДВ.В.ДЕС(), ДЕС.В.ДВ() и математическую функцию ОСНОВАНИЕ().
2. Можно написать небольшую программу на Питон.

Например, для задачи с таким условием:

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

1. Строится двоичная запись числа $4N$.

2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 10000 преобразуется в запись 100001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись является двоичной записью искомого числа R . Укажите такое **наименьшее** число N , для которого результат работы алгоритма больше 241. В ответе это число запишите в десятичной системе счисления.

Программа может выглядеть так:

```
for i in range(10,100):
```

```
    x=i*4
```

```
    y=bin(x)[2:]
```

```
    for j in range(2):
```

```
        if y.count('1')%2==0:
```

```
            y+='0'
```

```
        else:
```

```
            y+='1'
```

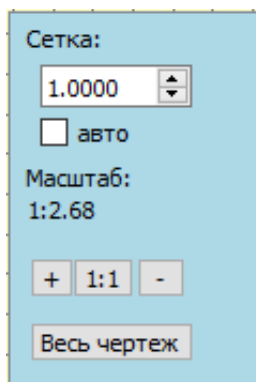
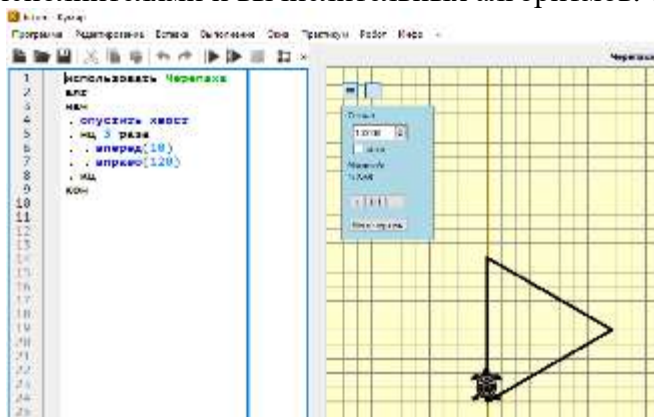
```
    r=int(y,2)
```

```
    if r>241:
```

```
        print(i)
```

```
        break
```

6 Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Эту задачу можно решить аналитически.



Можно решить в Кумире.

Важно выбрать масштаб 1:1 и убрать галочку в поле «авто»

Можно написать программу примерно следующего вида на Python:

```
from turtle import *
tracer(0)
r=20
for i in range(3):
    fd(10*r)
    rt(120)
up()
for x in range(-20,20):
    for y in range(-20,20):
        goto(x*r,y*r)
        dot(3,'red')
update()
```

Подробнее <https://www.youtube.com/live/yA0hN37nqWQ?si=xM1M4vkGa-41obw2> (Школково)

7 Умение определять объём памяти, необходимый для хранения графической и звуковой информации. **Важно** знать главную формулу информатики $2^i=N$. Можно посмотреть разбор по ссылке: <https://youtu.be/mXfkEe1sHjQ> (Алексей Кабанов)

8 Знание о методах измерения количества информации. *Задачи на комбинаторику или кодирование слов по алфавиту (последние устно решаются с помощью систем счисления или написанием программ с использованием перебора).*

Задачи на комбинаторику можно решить также с помощью кода. Например, такое условие: *Вася составляет 6-буквенные слова, в которых есть только буквы К, А, Н, Т, причём буква К используется в каждом слове ровно 2 раза. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?*

Решение:

```
s='КАНТ'
k=0
for a1 in s:
    for a2 in s:
        for a3 in s:
            for a4 in s:
                for a5 in s:
                    for a6 in s:
                        s1=a1+a2+a3+a4+a5+a6
                        if s1.count('К')==2:
                            k+=1
print(k)
```

Важно: если в условии задачи говорится, что кто-то составляет слова из слова, в котором есть повторяющиеся буквы (например, 'МОЛОКО'), то в переменную s надо задать комбинацию для алфавита без повторений ('МОЛК'). Обязательно следить за раскладкой клавиатуры, чтобы не перепутать русские и латинские буквы, начертание которых выглядит одинаковым, а коды – различны.

Другой вариант условия 8 задачи: Все четырёхбуквенные слова, в составе которых могут быть только буквы А, В, Т, О, Р, записаны в алфавитном порядке и пронумерованы, начиная с 1. Ниже приведено начало списка:

1. АААА
2. АААВ
3. АААО
4. АААР
5. АААТ
6. ААВА

Под каким номером в списке идёт слово ВАТА?

Решение:

```
s='АВОРТ'
k=0
for a1 in s:
    for a2 in s:
        for a3 in s:
            for a4 in s:
                s1=a1+a2+a3+a4
                k+=1
                if s1=='ВАТА':
                    print(k)
                    break
```

Эти задачи также можно решать с использованием библиотек.

Подробнее <https://youtu.be/xEIROiCJvg8> (Алексей Кабанов)

9 Умение обрабатывать числовую информацию в электронных таблицах

Надо знать основные функции ЭТ: МИН, МАКС, СРЗНАЧ, СУММ, СРЗНАЧЕСЛИ, СУММЕСЛИМН и т.д

Эту задачу также можно запрограммировать, но предварительно прилагаемый файл с расширением *xlsx* надо пересохранить в формате *csv*. Файл и программа должны храниться в одной папке.

Задача из Демо-2024: Откройте файл электронной таблицы, содержащей в каждой строке семь натуральных чисел. Определите количество строк таблицы, для чисел которых выполнены оба условия:

- в строке есть два числа, каждое из которых повторяется дважды, остальные три числа различны;
- среднее арифметическое всех повторяющихся чисел строки меньше среднего арифметического всех её чисел. В ответе запишите только число.

Решение:

```
f=open('9.csv')
k=0
for s in f:
    p=[]
    n=[]
    a=[int(i) for i in s.split(';')]
    for x in a:
        if a.count(x)==2:
            p.append(x)
```

```
if a.count(x)==1:
    n.append(x)
if len(p)==4 and len(n)==3 and sum(p)/4 < sum(a)/7:
    k+=1
print(k)
```

Можно посмотреть разбор https://youtu.be/TNv0d2T_E_E (Семён Овчинников)

10 Информационный поиск средствами операционной системы или текстового процессора
Может пригодиться задать Найти -> Параметры -> Только слова целиком

11 Умение подсчитывать информационный объём сообщения
Важно знать главную формулу информатики $2^i=N$

12 Умение анализировать результат исполнения алгоритма
Если в задаче используют Исполнитель Редактор, то можно написать программку на Питон

Пример из Демо-2022:

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 70 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (2222) ИЛИ нашлось (8888)

 ЕСЛИ нашлось (2222)

 ТО заменить (2222, 88)

 ИНАЧЕ заменить (8888, 22)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Решение:

```
s='8'*70
```

```
while '2222' in s or '8888' in s:
```

```
    if '2222' in s:
```

```
        s=s.replace('2222','88',1)
```

```
    else:
```

```
        s=s.replace('8888','22',1)
```

```
print(s)
```

Важно в методе replace задать третий параметр – 1, поскольку ищется первое вхождение подстроки в строке.

В некоторых задачах, известно какая строка будет получена на выходе и надо узнать, что подавали на вход. Например, Дана программа для редактора:

НАЧАЛО

ПОКА нашлось (01) ИЛИ нашлось (02) ИЛИ нашлось (03)

 заменить (01, 2302)

 заменить (02, 10)

 заменить (03, 201)

КОНЕЦ ПОКА

КОНЕЦ

Известно, что исходная строка начиналась с нуля, а далее содержала только единицы, двойки и тройки. После выполнения данной программы получилась строка, содержащая 40 единиц, 10 двоек и 8 троек. Сколько единиц было в исходной строке?

Решение:

```
for a in range(100):
    for b in range(100):
        for c in range(100):
            s='0'+a*'1'+b*'2'+c*'3'
            while '01' in s or '02' in s or '03' in s:
                s=s.replace('01','2302',1)
                s=s.replace('02','10',1)
                s=s.replace('03','201',1)
            if s.count('1')==40 and s.count('2')==10 and s.count('3')==8:
                print(a)
```

13 Умение использовать маску подсети.

Задача из Демо-2024: 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 маске сети.

Сеть задана IP-адресом 192.168.32.160 и маской сети 255.255.255.240.

Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса чётна? В ответе укажите только число.

Решение:

```
1 s=0
2 from ipaddress import *
3 for i in ip_network('192.168.32.160/255.255.255.240'):
4     if bin(int(i))[2:].count('1')%2==0:
5         print(i, ' ', bin(int(i))[2:])
6         s+=1
7 print(s)
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> [evaluate 13_1.py]
192.168.32.160 11000000101010000010000010100000
192.168.32.163 11000000101010000010000010100011
192.168.32.165 11000000101010000010000010100101
192.168.32.166 11000000101010000010000010100110
192.168.32.169 11000000101010000010000010101001
192.168.32.170 11000000101010000010000010101010
192.168.32.172 11000000101010000010000010101100
192.168.32.175 11000000101010000010000010101111
8
>>>
```

Задачи из сборника Крылова <https://youtu.be/CIrg1nxxiP8?si=wHsNfkBksL4S56XK>

14 Знание позиционных систем счисления

Важно знать формулы

число a^N в системе счисления с основанием a записывается как единица и N нулей: $a^N = \underbrace{10\dots0}_N$

число $a^N - a^M$ записывается в системе счисления с основанием a как $N-M$ старших цифр этой системы счисления, за которыми стоят M нулей: $a^N - a^M = \underbrace{(a-1)\dots(a-1)}_{N-M} \underbrace{0\dots0}_M$

Можно написать программу на Питон. Например, такое условие.

Значение выражения $343^5 - 7^9 + 48$ записали в системе счисления с основанием 7. Сколько цифр 6 содержится в этой записи?

Способ 1

```
r=343**5-7**9+48
s=[]
while r>0:
    d=r%7
    s.append(d)
    r=r//7
print(s.count(6))
```

Способ 2

```
r=343**5-7**9+48
k=0
while r>0:
    if r%7==6:
        k+=1
    r=r//7
print(k)
```

Задача из Демо 2024

Операнды арифметического выражения записаны в системе счисления с основанием 19.

$$98897x21_{19} + 2x923_{19}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 19-ричной системы счисления. Определите наибольшее значение x , при котором значение данного арифметического выражения кратно 18. Для найденного x вычислите частное от деления значения арифметического выражения на 18 и укажите его в ответе в десятичной системе счисления. Основание системы счисления указывать не нужно.

Решение:

Способ 1

```
alf='0123456789abcdefghi'
for x in alf:
    s=int(f'98897{x}21',19)+int(f'2{x}923',19)
    if s%18==0:
        print(s//18)
```

Способ 2

```
for x in range(19):
    s1=1+2*19+x*19**2+7*19**3+9*19**4+8*19**5+8*19**6+9*19**7
    s2=3+2*19+9*19**2+x*19**3+2*19**4
    s=s1+s2
    if s%18==0:
        print(s//18)
```

Важно знать законы алгебры логики, таблицы истинности логических операций, законы де Моргана и формулу $A \rightarrow B = \text{not } A + B$

Эти задачи бывают разных видов (отрезки, делители, конъюнкции). Решаются аналогичными способами

Задача из Демо 2024

Для какого наименьшего целого неотрицательного числа A выражение

$$(x + 2y < A) \vee (y > x) \vee (x > 60)$$

тождественно истинно, т.е. принимает значение 1 при любых целых неотрицательных x и y ?

Решение:

Используем флажок, который изначально делаем равным True. Если выражение окажется ложным хоть один раз, присваиваем флажку False.

```
for a in range(1000):
    f=True
    for x in range(100):
        for y in range(100):
            if ((x+2*y<a) or (y>x) or (x>60)) == 0:
                f=False
        if f:
            print(a)
            break
```

Если задача на конъюнкцию, при программировании используем оператор &

16 Вычисление рекуррентных выражений

Для решения этой задачи лучше написать программу (в комментариях приведено условие из Демо-2021)

#F(n) = 1 при n = 1;

#F(n) = n + F(n - 1), если n – чётно,

#F(n) = 2 × F(n - 2), если n > 1 и при этом n – нечётно.

#F(26)-?

Решение:

```
def f(n):
    if n==1:
        return 1
    elif n % 2 == 0:
        return n+f(n-1)
    else:
        return 2*f(n-2)
print(f(26))
```

Бывает, что при запуске программы на выполнение, она не выдаёт ответ («зависает»). Это происходит при большом количестве вызовов рекурсивной функции, когда в условии задачи используются большие числа.

Ситуацию можно попробовать исправить одним из способов:

1. *увеличением глубины рекурсии*

```
from sys import *  
setrecursionlimit(100000)
```

2. *подключение lru_cache*

```
from functools import lru_cache  
@lru_cache(None)
```

Пример использования lru_cache

Условие:

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$F(n) = n$, если $n \geq 2025$,

$F(n) = F(n + 1) - F(n + 2) + 7$, если $n < 2025$.

Чему равно значение выражения $F(15) - F(24)$?

Решение:

```
from functools import lru_cache  
@lru_cache(None)
```

```
def f(n):  
    if n >= 2025:  
        return n  
    return f(n+1)-f(n+2)+7
```

```
for i in range(2024, 14, -1):  
    f(i)
```

```
print(f(15)-f(24))
```

16-ую задачу также можно решать вручную или используя массивы.

Подробнее можно посмотреть разбор Алексея Кабанова

<https://www.youtube.com/watch?v=xw15lrAFt08&t=226s>

17 Умение создавать собственные программы (20–40 строк) для обработки целочисленной информации. Используется числовая последовательность из текстового файла

Задача из Демо 2024

В файле содержится последовательность натуральных чисел, каждое из которых не превышает 100000. Определите количество троек элементов последовательности, в которых ровно два из трёх элементов являются трёхзначными числами, а сумма элементов тройки не больше максимального элемента последовательности, оканчивающегося на 13.

Гарантируется, что в последовательности есть хотя бы одно число, оканчивающееся на 13. В ответе запишите количество найденных троек чисел, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Решение:

```
f=open('17.txt')
a=[int(s) for s in f]
mx=max([s for s in a if s%100==13])
ms=-100000
k=0
for i in range(len(a)-2):
    k3=0
    if 99<a[i]<1000: k3+=1
    if 99<a[i+1]<1000: k3+=1
    if 99<a[i+2]<1000: k3+=1
    if k3==2:
        s=a[i]+a[i+1]+a[i+2]
        if s<=mx:
            k+=1
            ms=max(ms,s)
print(k,ms)
```

В приведённом решении все числа были считаны из файла в массив **a**.

Важно:

- текстовый файл с исходными данными и программа должны быть сохранены в одной папке, если при открытии файла не используется полный путь к файлу
- в цикле для обработки данных (при обработке пар чисел с индексами **i** и **i+1**) следить за тем, чтобы не выйти за границы массива
- если в файле даны отрицательные и положительные числа, то при проверке на какую цифру оканчивается число, необходимо использовать модуль этого числа ($\text{abs}(x)$)

18 Умение использовать электронные таблицы для обработки целочисленных данных. Чаще всего это задача про Робота, который перемещается по лабиринту. *Задача решается в ЭТ*

19-21 Умение анализировать алгоритм логической игры и найти выигрышную стратегию. Существуют разные способы решения:

1) Можно решать вручную. Как это делать можно посмотреть у Семёна Овчинникова

<https://youtu.be/MzygE1NdUCA?si=vRXtQ5FAVaD2ig2n>

2) Данная задача может быть решена в электронных таблицах. Метод, предложенный Артёмом Имаевым, можно посмотреть по ссылке: <https://youtu.be/iQ8OFqML2b0>

3) Кроме того, существуют различные алгоритмы программ.

a. Семён Овчинников <https://youtu.be/waScF4QAnK0>

b. Алексей Кабанов <https://youtu.be/saqgoCS6wjK>

Пример задачи (5 вариант из сборника Крылова за 2023 год)

Решение методом Овчинникова С.

19

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Например, имея кучу из 19 камней, за один ход можно получить кучу из 20 или 38 камней. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 181. Победителем считается игрок, сделавший последний ход, т. е. первым получивший кучу, в которой будет 181 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 180$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т. е. не являющиеся выигрышными независимо от игры противника. Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

20

Для игры, описанной в задании 19, найдите два наименьших 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 того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

21

Для игры, описанной в задании 19, найдите минимальное значение S , при котором одновременно выполняются два условия:

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

Решение:

```
print('zadacha 19')
def f(x,y,p):
    if x+y >= 142 or p>3:
        return p==3
    return f(x+2,y,p+1) or f(x*2,y,p+1) or f(x,y+2,p+1) or f(x,y*2,p+1)
for s in range(1,139):
    if f(2,s,1):
        print(s)
        break

print('zadacha 20')
def f(x,y,p):
    if x+y >= 142 or p>4:
        return p==4
    if p%2!=0:
        return f(x+2,y,p+1) or f(x*2,y,p+1) or f(x,y+2,p+1) or f(x,y*2,p+1)
    else:
        return f(x+2,y,p+1) and f(x*2,y,p+1) and f(x,y+2,p+1) and f(x,y*2,p+1)
```

```

for s in range(1,139):
    if f(2,s,1):
        print(s)

print('zadacha 21')
def f(x,y,p):
    if x+y >= 142 or p>5: # для проверки написать p>3 и число, которое выдаст не подходит
        return p==3 or p==5
    if p%2==0:
        return f(x+2,y,p+1) or f(x*2,y,p+1) or f(x,y+2,p+1) or f(x,y*2,p+1)
    else:
        return f(x+2,y,p+1) and f(x*2,y,p+1) and f(x,y+2,p+1) and f(x,y*2,p+1)
for s in range(1,139):
    if f(2,s,1):
        print(s)

```

Кабанов А.

(№ 3075) (А. Кабанов) Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу два камня или увеличить количество камней в куче в два раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 25. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 25 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 24$.

Ответьте на следующие вопросы:

Вопрос 1. Найдите минимальное значение S , при котором Ваня выигрывает своим первым ходом при любой игре Пети. $m=2$

Вопрос 2. Сколько существует значений S , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход; нет $m=1$
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. если $m=3$

Вопрос 3. Найдите два значения S , при которых одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. нет $m=2$

Найденные значения запишите в ответе в порядке возрастания. если $m=4$

```

def f(s,m):
    if s>=25: return m%2==0
    if m==0: return 0
    h=[f(s+2,m-1), f(s*2,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print('19 ',[s for s in range(1,25) if f(s,2)])
print('20 ',[s for s in range(1,25) if not f(s,1) and f(s,3)])
print('21 ',[s for s in range(1,25) if not f(s,2) and f(s,4)])

```

22 Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы.
Задачу можно решить в ЭТ. Может очень пригодиться умение использовать функцию ВПР при большом количестве процессов.

23 Умение анализировать результат исполнения алгоритма

Эту задачу можно решать различными способами.

- 1) Можно делать письменно на листке бумаги
- 2) Можно написать рекурсивный алгоритм
- 3) Можно написать вспомогательную программу с использованием массива

Например:

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Умножить на 2

Сколько существует программ, для которых при исходном числе 1 результатом является число 20, и при этом траектория вычислений содержит число 10?

Решение с использованием рекурсии

```
def f(x,y):
    if x>y:
        return 0
    elif x==y:
        return 1
    else:
        return f(x+1,y)+f(x*2,y)
print(f(1,10)*f(10,20))
```

Решение с использованием массива

```
a=[0]*25
a[1]=1
a[2]=2
for i in range(3,11):
    if i % 2 == 0:
        a[i]=a[i-1] + a[i // 2]
    else:
        a[i]=a[i-1]
for i in range(11,20):
    a[i]=a[10]
a[20]=a[19] + a[10]
print(a[20])
```

Можно решить в электронных таблицах. На рисунке показано решение для задачи с таким условием: *Исполнитель преобразует число на экране. У исполнителя есть три команды, которым присвоены номера:*

1. Прибавить 1
2. Умножить на 2
3. Умножить на 3

Сколько существует программ, которые преобразуют исходное число 2 в число 36, и при этом траектория вычислений содержит число 12 и не содержит числа 30?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
1		2	3	4	5	6	7	8	9	10	11	12															
2	+1		1	1	2	2	4	4	6	7	9	9															
3	*2			1		1		2		2		4															
4	*3					1			1			2															
5		1	1	2	2	4	4	6	7	9	9	15															
6																											
7																											
8		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
9	+1		15	15	15	15	15	15	15	15	15	15	15	15	30	30	45	45	60	60	0	0	15	15	30	30	
10	*2												15	15	15	15	15	15	15	15	15	15	15	15	15	15	
11	*3																									15	
12		15	15	15	15	15	15	15	15	15	15	15	15	30	30	45	45	60	60	0	0	15	15	30	30	60	
13																											

24 Умение создавать собственные программы (10–20 строк) для обработки символьной информации

Задачи вида

<p>Определите сколько строк в файле, в которых символов А больше чем Е</p>	<p>Определите длину максимальной последовательности, в которой символы равны друг другу</p>	<p>Определите максимальное количество идущих подряд символов, среди которых не более одной буквы А</p>
<pre>f=open('24.txt') k=0 for line in f.readlines(): if line.count('A')>line.count('E'): k+=1 print(k)</pre>	<pre>f=open('24.txt') s=f.readline() k,m=1,0 for i in range(1,len(s)): if s[i-1]==s[i]: k+=1 m=max(m,k) else: k=1 print(m)</pre>	<pre>f=open('24.txt') s=f.read() s=s.split('A') m=0 for i in range(len(s)-1): m=max(m,len(s[i])+len(s[i+1])) print(m+1)</pre>

Задача из Демо-2022: Текстовый файл состоит из символов P, Q, R и S. Определите максимальное количество идущих подряд символов в прилагаемом файле, среди которых нет идущих подряд символов P.

```
f=open('24.txt')
s=f.read()
k=1
m=1
for i in range(len(s)-1):
```

```

if s[i]!='P' or s[i+1]!='P':
    k+=1
else:
    m=max(m,k)
    k=1
print(m)

```

Задача из Демо-2024: Текстовый файл состоит из символов T, U, V, W, X, Y и Z. Определите в прилагаемом файле максимальное количество идущих подряд символов (длину непрерывной подпоследовательности), среди которых символ T встречается ровно 100 раз. Для выполнения этого задания следует написать программу.

```

s=open('24.txt').readline()
s='T'+s+'T'
r=100
a = []
for i in range(len(s)):
    if s[i]=='T':
        a+=[i]
m=0
for i in range(len(a)-r-1):
    l=abs(a[i+r+1]-a[i])-1
    m =max(m,l)
print(m)

```

Важно:

- текстовый файл с исходными данными и программа должны быть сохранены в одной папке, если при открытии файла не используется полный путь к файлу
- в цикле для обработки строки (при обработке элементов с индексами **i** и **i+1**) следить за тем, чтобы не выйти за границы строки

25 Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации

Задание из Демо-2022: Пусть M – сумма минимального и максимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то значение M считается равным нулю. Напишите программу, которая перебирает целые числа, большие 700 000, в порядке возрастания и ищет среди них такие, для которых значение M оканчивается на 8. Выведите первые пять найденных чисел и соответствующие им значения M . Формат вывода: для каждого из пяти таких найденных чисел в отдельной строке сначала выводится само число, затем – значение M . Строки выводятся в порядке возрастания найденных чисел.

Решение:

```

k=1
ch=700001
while k<=5:
    dmin=0
    dmax=0
    for d in range(2,int(ch**0.5)+1):
        if ch%d==0:

```



```

dmin=d
dmax=ch//dmin
break
m=dmin+dmax
if m%10==8:
    print(ch,m)
    k+=1
ch+=1

```

Задание из Демо-2024: Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:

- символ «?» означает ровно одну произвольную цифру;
- символ «*» означает любую последовательность цифр произвольной длины; в том числе «*» может задавать и пустую последовательность.

Например, маске $123*4?5$ соответствуют числа 123405 и 12300405 . Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1?2157*4$, делящиеся на 2024 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 2024 .

Решение:

```

from fnmatch import fnmatch
for x in range(0,10**10+1,2024):
    if fnmatch(str(x),'1?2157*4'):
        print(x,x//2024)

```

26 Умение обрабатывать целочисленную информацию с использованием сортировки

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов. Известно, какой объём занимает файл каждого пользователя. По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Входные данные.

В первой строке входного файла находятся два числа: S – размер свободного места на диске (натуральное число, не превышающее $10\,000$) и N – количество пользователей (натуральное число, не превышающее 1000). В следующих N строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

```

f=open('26.txt')
s,n=map(int,f.readline().split())
a=[]
for i in range(n):
    a.append(int(f.readline()))
a.sort()
sum,i=0,0
while sum<=s:
    sum+=a[i]

```



```
i+=1
i-=1
sum-=a[i]+a[i-1]
m=i
while sum+a[m]<=s:
    m+=1
print(i,a[m-1])
```

Эта задача, как правило, также решается в электронных таблицах

27 Умение создавать собственные программы (20–40 строк) для анализа числовых последовательностей

Для задачи А возможно переборное решение.

На вход программы поступает последовательность из N целых положительных чисел. Рассматриваются все пары различных элементов последовательности. Необходимо определить количество пар чисел, разность которых кратна 13, а произведение чётно.

Входные данные.

Даны два входных файла (файл А и файл В), каждый из которых содержит в первой строке количество чисел N ($2 \leq N \leq 12000$). В каждой из последующих N строк записано одно целое положительное число, не превышающее 10 000.

Программа должна вывести в первой строке одно число: количество пар чисел, разность которых кратна 13, а произведение чётно. Если подходящих пар нет, нужно вывести "NO".

```
f=open('27-A.txt')
n=int(f.readline())
k=0
a=[]
for i in range(n):
    a.append(int(f.readline()))
for i in range(n-1):
    for j in range(i+1,n):
        if (a[i]-a[j]) % 13 == 0 and (a[i]*a[j]) % 2 ==0:
            k+=1
print(k)
```

Для файла В нужно использовать оптимальный алгоритм

**УДАЧИ
И
ВЫСОКИХ БАЛЛОВ
НА ЕГЭ!**

