

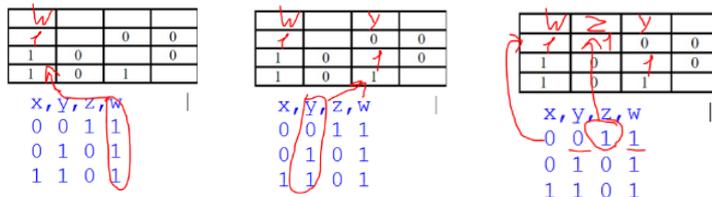
Задача 2. Демо-24

значо	ПИТО
К	Н
\vee	or
\wedge	and
\rightarrow	<=
\equiv	==
\neg	not

		0	0
1	0		0
1	0	1	

x, y, z, w
 0 0 1 1
 0 1 0 1
 1 1 0 1

```
print('x,y,z,w')
for x in 0,1:
    for y in 0,1:
        for z in 0,1:
            for w in 0,1:
                if ((x and (not y)) or (y==z) or (not w))==0:
                    print(x,y,z,w)
```



=ВПР(D1034;Товар!A:F;5;0)

ВПР(искомое_значение; таблица; номер_столбца; [интервальный_просмотр])

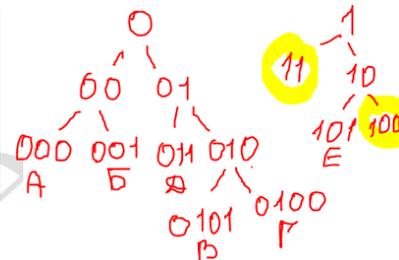
Задача 3. Демо-24

1. Искомое значение – это id товара.
2. Указываем лист откуда берем значение
3. Указываем столбец из таблицы, который нужно вернуть
4. Пишем 0 – не сортировать

Задача 4. Демо-2024

1. Обращаем внимание сколько всего букв.
2. Строим дерево.
3. Определяем свободные ветки
4. Если кодируется весь алфавит – оставляем одну ветку

А	000
Б	001
В	0101
Г	0100
Д	011
Е	101



Задача 5. Демо-2024

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

1. Строится двоичная запись числа N .
2. Далее эта запись обрабатывается по следующему правилу:
 - а) если число N делится на 3, то к этой записи дописываются три последние двоичные цифры;
 - б) если число N на 3 не делится, то остаток от деления умножается на 3, переводится в двоичную запись и дописывается в конец числа.

Полученная таким образом запись является двоичной записью искомого числа R .

3. Результат переводится в десятичную систему и выводится на экран. Например, для исходного числа $12 = 1100_2$, результатом является число $1100100_2 = 100$, а для исходного числа $4 = 100_2$ это число $10011_2 = 19$.

Укажите **минимальное** число R , большее 151, которое может быть получено с помощью описанного алгоритма. В ответе запишите это число в десятичной системе счисления.

```
for n in range(1,50):
    r = bin(n)[2:]
    if n%3==0:
        r = r + r[-3:]
    else:
        r = r+bin((n%3)*3)[2:]
    print(n,int(r,2))
```

```
def alg(n):
    s = ''
    while n > 0:
        s = str(n%3)+s
        n = n//3
    return s
for n in range(2,100):
    s = alg(n)
    if n%3 == 0:
        s = s + s[-2:]
    else:
        s = s + alg((n%3)*3)
    print(n,int(s,3))
```

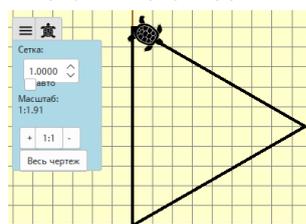
```
for n in range(1, 50):
    r = bin(n)[2:]
    if r[-1] == '0':
        r = '10'+ r
    else:
        r = '1'+ r + '01'
    print(n, int(r, 2))
```

Задача 6. Демо-2024

Черепахе был дан для исполнения следующий алгоритм:

Повтори 7 [Вперёд 10 Направо 120].

Определите, сколько точек с целочисленными координатами будут находиться внутри области, которая ограничена линией, заданной этим алгоритмом. Точки на линии учитывать не следует.



использовать **Черепаха**

- алг
 нач
 . опустить хвост
 . нц 7 раза
 . . вперед(10)
 . . вправо(120)
 . кц
 кон

Формула нахождения объема звуковой записи.

$$V = N * F * B * t, \text{ где}$$

- V — объем файла;
- N — количество волн;
- F — частота дискретизации (Гц);
- B — разрешение / глубина кодирования;
- t — время записи в секундах.

Задача 7. Демо-2024

Прибор автоматической фиксации нарушений правил дорожного движения делает цветные фотографии размером 1024×768 пикселей, используя палитру из 4096 цветов. Для передачи снимки группируются в пакеты по 256 штук. Определите размер одного пакета фотографий в Мбайт. В ответе запишите только число.

- 1) Находим $i=12$ т. к. $2^{12}=4096$
 - 2) $i = k * i = (1024 * 768 * 12) / 8 / 1024 / 1024 = 1,125$ Мбайт
- Для перевода в Мбайт разделили на 8, на 1024 и еще раз на 1024
- 3) Пакет состоит из 256 таких картинок $i = 1,125 * 256 = 288$ Мбайт

$$V = N * i, \text{ где}$$

- V — объем памяти, требуемый для хранения изображения;
- N — размер (произведение ширины на высоту) изображения в пикселях;
- i — глубина кодирования цвета.

$$N = 2^B, \text{ где}$$

- N — количество уровней дискретизации;
- B — объем одного измерения.

Задача 8. Демо-2023

```
a='01234567'
a1='1234567'
c=0
for i in a1:
    for q in a:
        for w in a:
            for e in a:
                for r in a:
                    s=i+q+w+e+r
                    if s.count('6')==1:
                        if ('16' not in s) and ('61' not in s):
                            if ('36' not in s) and ('63' not in s):
                                if ('56' not in s) and ('65' not in s):
                                    if ('76' not in s) and ('67' not in s):
                                        c+=1
print(c)
```

Все четырёхбуквенные слова, в составе которых могут быть только буквы Л, Е, М, У, Р, записаны в алфавитном порядке и пронумерованы, начиная с 1.

Ниже приведено начало списка.

1. EEEE
2. EEEЛ
3. EEEМ
4. EEEР
5. EEEУ
6. EELE
- ...

Под каким номером в списке идёт первое слово, которое начинается с буквы Л?

Какое слово будет под №415(ГРИНЧ)

```
s='грнрч'
s1=0
for q in s:
    for w in s:
        for e in s:
            for r in s:
                for t in s:
                    slovo=q+w+e+r+t
                    s1+=1
                    if s1==415:
                        print(slovo)
```

```
126 леее s='елмру'
127 леел s1=0
128 леем for q in s:
129 леер for w in s:
130 лееу for e in s:
131 леле for r in s:
132 лелл slovo=q+w+e+r
133 лелм s1+=1
134 лелр if q=='л':
135 лелу print(s1,slovo)
```

Даша составляет 7-буквенные слова из букв БАНДЕРОЛЬ. Мягкий знак используется в каждом слове не больше одного раза. Остальные буквы могут быть использованы сколько угодно раз или не использоваться совсем.

Сколько слов может составить Даша, если известно, что буква Е не может стоять рядом с согласными?

```
from itertools import*
s = sorted('01234567')
c=0
for i in product(s,repeat=6):
    x=''.join(i)
    if len(set(x))==6 and x[0]!='0' and int(x,8)%5==0:
        for i in '0246':
            x=x.replace(i,'*')
        for i in '1357':
            x=x.replace(i,'+')
        if ('++' not in x) and('**' not in x):
            c+=1
print(c)
```

```
from itertools import*
k = 0
s = sorted('бандероль')
for i in product(s,repeat=7):
    x=''.join(i)
    if x.count('ь')<=1:
        for j in 'бндрл':
            x=x.replace(j,'*')
        if x.count('*e')==0 and x.count('e*')==0:
            k+=1
print(k)
```

Сколько существует восьмеричных шестизначных чисел, в которых все цифры различны, никакие две чётные или две нечётные цифры не стоят рядом и десятичная запись которых делится на 5?

Задача 11. Демо-2024

При регистрации в компьютерной системе каждому объекту присваивается идентификатор, состоящий из 60 символов и содержащий только десятичные цифры и символы из 250-символьного специального алфавита. В базе данных для хранения каждого идентификатора отведено одинаковое и минимально возможное целое число байт. При этом используется посимвольное кодирование идентификаторов, все символы кодируются одинаковым и минимально возможным количеством бит. Определите объём памяти (в Кбайт), необходимый для хранения 65 536 идентификаторов. В ответе запишите только целое число – количество Кбайт.

- 1) Определяем мощность алфавита $N=10$ цифр +250 сим=260
- 2) Определяем $260 \leq 2^i$, $i=9$
- 3) Определяем Объем 1 идентификатора $N = 2^i$
 $I = K \cdot i = 60 \cdot 9 = 540$ бит
- 4) Переводим биты в байты
 $540 : 8 = 67,5$
- 5) Округляем до целого числа байт 68
- 6) Определяем общий объем в Кбайт
 $I = (65536 \cdot 68) / 1024 = 4352$

$N = 2^i$
 $V = N \cdot i$, где
 V — объём сообщения;
 N — количество символов;
 i — объём символа.

Задача 12. Демо-2024

Дана программа для Редактора:

```
НАЧАЛО
ПОКА нашлось (52) ИЛИ нашлось (2222) ИЛИ нашлось (1122)
    ЕСЛИ нашлось (52)
        ТО заменить (52, 11)
    КОНЕЦ ЕСЛИ
    ЕСЛИ нашлось (2222)
        ТО заменить (2222, 5)
    КОНЕЦ ЕСЛИ
    ЕСЛИ нашлось (1122)
        ТО заменить (1122, 25)
    КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```

На вход приведённой выше программе поступает строка, начинающаяся с цифры «5», а затем содержащая n цифр «2» ($3 < n < 10\,000$). Определите наибольшее значение n , при котором сумма цифр в строке, получившейся в результате выполнения программы, равна 64.

```
max_n=0
for n in range(3,1000):
    s = '5'+ '2'*n
    while ('52' in s) or ('2222' in s) or ('1122' in s):
        s=s.replace('52', '11',1)
        s=s.replace('2222', '5',1)
        s=s.replace('1122', '25',1)
    if s.count('5')*5+s.count('2')*2+s.count('1')==64:
        max_n=n
print(max_n)
```

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки символов.

1. заменить (v, w)
2. нашлось (v)

Первая команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Если цепочки v в строке нет, эта команда не изменяет строку. Вторая команда проверяет, встречается ли цепочка v в строке исполнителя Редактор.

Дана программа для исполнителя Редактор:

```

НАЧАЛО
ПОКА нашлось(8888)
    заменить(8888, 44)
    заменить(4444, 848)
КОНЕЦ ПОКА
КОНЕЦ

```

На вход приведённой выше программе поступает строка, содержащая l цифр «8» ($3 < l < 1000$). Определите наименьшее возможное количество цифр «8» в строке, при котором сумма цифр строки, получившейся в результате выполнения программы, равна не менее 52.

Дана программа для Редактора:

```

НАЧАЛО
ПОКА нашлось (1111) ИЛИ нашлось (7777)
    ЕСЛИ нашлось (1111)
        ТО заменить (1111, 77)
        ИНАЧЕ заменить (7777, 1)
    КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ

```

На вход приведённой выше программе поступает строка, начинающаяся с цифры «7», а затем содержащая n цифр «1» ($3 < n < 10000$).

Определите **наибольшее** возможное значение суммы цифр в строке, которая может быть результатом выполнения программы.

Задание 13.

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.

Сеть задана IP-адресом 105.224.200.224 и сетевой маской 255.255.255.224. Сколько в этой сети IP-адресов, для которых количество единиц в двоичной записи IP-адреса кратно 4? В ответе укажите только число.

Сеть задана IP-адресом одного из входящих в неё узлов 11.92.135.56 и сетевой маской 255.224.0.0.

Найдите в данной сети наибольший IP-адрес, который может быть назначен компьютеру. В ответе укажите найденный IP-адрес без разделителей.

Задание 14. 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.

Значение арифметического выражения

$$3 \cdot 3125^8 + 2 \cdot 625^7 - 4 \cdot 625^6 + 3 \cdot 125^5 - 2 \cdot 25^4 - 2024$$

записали в системе счисления с основанием 25. Сколько значащих нулей содержится в этой записи?

Задание 15. 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 ?

Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наибольшего натурального числа A логическое выражение

$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 28) \rightarrow \neg \text{ДЕЛ}(x, 49))$$

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

minimum = 200000

def func(x, A1, A2):

return (130 <= x <= 171) <= (((150 <= x <= 185) and (not(A1 <= x <= A2))) <= (not(130 <= x <= 171)))

for A1 in range(1, 250): #цикл начала промежутка

for A2 in range(A1+1, 250): #цикл конца промежутка

proverka = True #переменная для проверки истинности функции

for x in range(1, 1000):

if func(x, A1, A2) == False:

proverka = False

break

if proverka == True:

minimum = min(minimum, A2-A1)

print(minimum)

Задача 16. Демо-2024

Алгоритм вычисления значения функции $F(n)$, где

задан следующими соотношениями:

```

def F(n):
    if n > 2024: return n

```

```

for n in range(3, 1000):
    s = '8'*n
    while '8888' in s:
        s=s.replace('8888', '44',1)
        s=s.replace('4444', '848',1)
    if s.count('8')*8+s.count('4')*4 <=52:
        print(len(s))

```

```

max_sum=0
for n in range(4, 10000):
    s='7'+ '1'*n
    while '1111' in s or '7777' in s:
        if '1111' in s:
            s=s.replace('1111', '77',1)
        else:
            s=s.replace('7777', '1',1)
    if s.count('7')*7+s.count('1')>max_sum:
        max_sum=s.count('7')*7+s.count('1')
    print(max_sum)

```

```

from ipaddress import *
count=0
ip_net = ip_network("105.224.200.224/255.255.255.224")
for ip_ad in ip_net:
    s=bin(int(ip_ad))[2:]
    if s.count('1')%4==0:
        count+=1
print(count)

from ipaddress import *
max_ip=""
predposledni=""
ip_net = ip_network("11.92.135.56/255.224.0.0", 0)
for ip_ad in ip_net:
    predposledni= max_ip
    max_ip= ip_ad
print(predposledni)

```

```

s='0123456789abcdefghi'
for x in s:
    s1=int('98897'+x+'21', 19)+int('2'+x+'923', 19)
    if s1%18==0:
        max_kr=x
print((int('98897'+max_kr+'21', 19)+int('2'+max_kr+'923', 19))/18)

n=3*3125**8+2*625**7-4*625**6+3*125**5-2*25**4-2024
count=0
while n>0:
    if n%25==0:
        count+=1
    n//=25
print(count)

```

```

for a in range(0, 1001):
    f=0
    for x in range(0, 1000):
        for y in range(0, 1000):
            if ((x+2*y < a) or (y>x) or (x>60)) == False:
                f=1
                break
    if f==0:
        print(a)

```

```

for a in range(1, 1001):
    f=0
    for x in range(1, 1000):
        if ((x%a!=0) <= ((x%28==0) <= (x%49!=0))) == False:
            f=1
            break
    if f==0:
        print(a)

```

На числовой прямой даны два отрезка: $B = [24, 900]$ и $C = [47, 115]$. Укажите наименьшую возможную длину такого отрезка A , для которого логическое выражение

$$A \rightarrow B = \overline{A} \vee B \quad (x \notin C) \vee ((x \notin A) \wedge (x \in B)) \vee (x \in C)$$

$$(x \notin C) \vee (x \in A) \vee (x \notin B)$$



```

from sys import *
setrecursionlimit(10000)
def f(n):

```

Досрок1-2024 *

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число
 $F(n) = 1$ при $n < 7$;
 $F(n) = n + 2 + F(n - 1)$, если $n \geq 7$.
 Чему равно значение выражения $F(2024) - F(2020)$?

```
F=[1]*3000
for n in range(2,3000):
    if n>=7:
        F[n]=n+2+F[n-1]
    else:
        F[n]=1
print(F[2024]-F[2020])
```

Задача 17. Демо-2024. Сохрани программу в ту же папку где файл 17.txt

В файле содержится последовательность натуральных чисел, каждое из которых не превышает 100 000. 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.

```
f=open('17_2024.txt')
a=[int(i) for i in f]
mx=-100500
c=0
s=0
for i in range(len(a)):
    if a[i]>mx and a[i]%100==13:
        mx=a[i]
for i in range(len(a)-2):
    if ((len(str(a[i]))==3)+(len(str(a[i+1]))==3)+(len(str(a[i+2]))==3))==2:
        if a[i]+a[i+1]+a[i+2]<mx:
            c+=1
            s=max(s,a[i]+a[i+1]+a[i+2])
print(f=open('17_2022.txt')
a=[int(i) for i in f]
mx=-100500
c=0
ms=-100500
for i in range(len(a)):
    if a[i]>mx and a[i]%3==0:
        mx=a[i]
for i in range(len(a)-1):
    if (abs(a[i])%3==0 or abs(a[i+1])%3==0) and ((a[i]+a[i+1])<=mx):
        c+=1
        ms=max(ms,a[i]+a[i+1])
print(c,ms)
```

Задача 17. Демо-2022

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар последовательности, в которых хотя бы одно число делится на 3, а сумма элементов пары не более максимального элемента последовательности, кратного 3. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

```
f=open('17_2022.txt')
a=[int(i) for i in f]
mx=-100500
c=0
ms=-100500
for i in range(len(a)):
    if a[i]>mx and a[i]%3==0:
        mx=a[i]
for i in range(len(a)-1):
    if (abs(a[i])%3==0 or abs(a[i+1])%3==0) and ((a[i]+a[i+1])<=mx):
        c+=1
        ms=max(ms,a[i]+a[i+1])
print(c,ms)
```

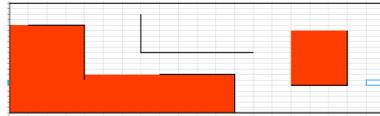
Задача 18. Демо-2024

Квадрат разбит на $N \times N$ клеток ($1 < N < 30$). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: **вправо** или **вниз**. По команде **вправо** Робот перемещается в соседнюю правую клетку, по команде **вниз** – в соседнюю нижнюю. Квадрат ограничен внешними стенами. Между соседними клетками квадрата также могут быть внутренние стены. Сквозь стену Робот пройти не может. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клеткам маршрута Робота. В «куловых» клетках поля – тех, которые справа и снизу ограничены стенами, Робот не может продолжать движение, поэтому накопленная сумма считается итоговой. Таких конечных клеток на поле может быть несколько, включая правую нижнюю клетку поля. При разных запусках итоговые накопленные суммы могут различаться. Определите максимальную и минимальную денежные суммы, среди всех возможных итоговых сумм, которые может собрать Робот, пройдя из левой верхней клетки в конечную клетку маршрута. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу размером $N \times M$, каждая ячейка которой соответствует клетке квадрата. Внутренние и внешние стены обозначены утолщенными линиями.

99	77	66	43	73	71	28	58	9	50	80	44	75	26	27	14	56	76	29	17
17	20	60	74	67	58	8	75	44	12	15	12	33	39	67	35	71	14	49	12
26	7	48	25	22	53	33	58	75	76	30	73	8	91	78	81	37	75	25	17
70	72	36	26	14	79	6	11	59	25	80	36	11	61	76	50	88	74	11	19
74	77	65	68	14	31	48	45	66	45	39	51	52	69	61	37	48	64	28	24
20	49	31	6	23	63	18	13	13	8	9	19	18	67	46	50	79	41	37	12
18	75	64	43	59	12	16	18	14	8	19	8	8	40	59	81	50	41	21	18
50	67	67	45	48	48	15	8	14	12	9	7	9	70	56	50	81	41	45	13
13	60	60	79	67	63	11	5	13	12	13	20	11	39	60	62	35	71	14	8
92	55	25	71	39	37	30	19	19	12	9	8	32	32	26	16	37	78	48	23
74	74	92	82	38	40	12	6	9	11	18	20	5	55	52	65	49	17	63	14
50	42	48	78	99	41	11	15	14	6	15	11	8	28	70	50	75	28	41	25
83	82	30	71	71	56	20	7	18	5	10	16	18	11	14	52	59	41	41	23
36	38	61	28	38	78	15	62	15	31	62	42	29	35	25	36	52	19	64	16
42	43	94	89	81	62	17	22	44	65	22	63	11	74	33	50	28	17		
61	32	25	85	42	79	68	64	36	25	53	54	46	50	55	71	62	44	42	23
51	77	100	52	100	16	50	41	62	45	20	13	79	69	63	70	51	23	36	7
61	37	92	99	48	38	17	45	48	54	73	63	64	66	62	32	9	77	60	73
33	36	88	84	73	76	38	78	16	76	36	17	16	19	13	33	20	16	37	28
24	73	72	79	49	28	16	73	14	20	44	44	49	12	36	71	70	26	19	14

- Копируем таблицу ниже
- Удаляем все числа в скопированной таблице, кроме левой верхней ячейки.
- Красным выделил ячейки туловищевые для робота.



- Заполняем формулами горизонтальные и вертикальные пути.



- Вписываем формулу поиска максимального пути и растягиваем на все оставшиеся ячейки

99	176	242	285	338	429	457	515	524	574	654	698	773	799	826	840	896	972	1001	1018	
116	196	302	376	443	501	509	590	634	646	669	710	806	845	912	947	1018	1032	1101	1113	
142	203	350	401	465	554	588	648	723	799	829	902	910	1001	1079	1160	1197	1272	1297	1314	
212	291	330	356	479	633	639	659	782	824	909	945	956	1062	1155	1210	1298	1372	1383	1396	
493	664	704	704	848	893	948	999	1051	1131	1216	1253	1346	1436	1464	1488	1501	1513			
516	727	748	747	861	901	957	1018	1069	1198	1262										
575	739	753	735	875	909	976	1026	1077	1238	1321										
623	787	802	743	889	921	985	1033	1086	1308	1377										
690	850	861	748	761	780	793	813	824	1347	1437										
729	887	907	920	939	951	960	968	980	1369	1463										
787	927	939	945	954	965	963	1003	1008	1424	1515										
886	988	999	1014	1028	1034	1051	1066	1074	1452	1585										
957	1013	1033	1040	1058	1063	1073	1089	1107	1463	1599										
1136	1518	1624																		
1168	1581	1635																		
1214	1631	1690	1761	1823	1867	1909	1930													
1293	1700	1763	1833	1884	1907	1943	1950													
1357	1766	1828	1865	1893	1984	2044	2115													
1373	1785	1841	1898	1918	2000	2081	2153													
1422	1797	1877	1969	2039	2095	2113	2167													

- Нажимаем сочетание клавиш CTRL-N и заменяем макс на мин.

Задача 19-21. Демо-2024

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**. 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 129. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу из 129 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 128$. Будем говорить, что игрок имеет **выигрышную стратегию**, если он может выиграть при любых ходах противника.

Укажите минимальное значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

$\Pi 1 \quad 32 \times 2 = 64 \quad 63 + 1 = 64$
 $B 1 \quad 64 + 1 = 65$
 $\Pi 2 \quad 65 \times 2 = 130$

$+ 1 \quad \times 2 \quad [129]$
 $\Pi \quad 64 + 1 = 65$
 $B \quad 65 \times 2 = 130$
 $\uparrow \quad 129 : 2 = 64,5 \approx 65$
 $2) \quad 65 - 1 = (64)$

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.

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.

Если найдено несколько значений S , в ответе запишите минимальное из них.

$\Pi 1$ $62+1=63$ $62 \times 2=124$
 $B1$ $63+1=64$ $124 \times 2=248$
 $\Pi 2$ $64+1=65$
 $B2$ $65 \times 2=130$

62

```
def f(x, p):
    if x <= 19 or p > 2:
        return p == 2 or p == 4
    if p % 2 == 0:
        return f(x-2, p+1) or f(x-5, p+1) or f(x//3, p+1)
    else:
        return f(x-2, p+1) and f(x-5, p+1) and f(x//3, p+1)
print([s for s in range(100, 19, -1) if (f(s, 0))])
```

```
def f(x, c, pob):
    if x >= 129:
        return c in pob
    if c > max(pob):
        return 0
    moves = [f(x+1, c+1, pob), f(x*2, c+1, pob)]
    if c % 2 != max(pob) % 2:
        return any(moves)
    else:
        return all(moves)

for s in range(1, 128 + 1):
    if f(s, 0, [2]) == 1:
        print('№ 19:', s)
    if f(s, 0, [3]) == 1:
        print('№ 20:', s)
    if f(s, 0, [2, 4]) == 1 and f(s, 0, [2]) == 0:
        print('№ 21:', s)
```

две кучи

```
def f(x1, x2, c, pob):
    if x1 + x2 >= 77:
        return c in pob
    if c > max(pob):
        return 0
    moves = [f(x1+1, x2, c+1, pob), f(x1, x2+1, c+1, pob), f(x1*2, x2, c+1, pob), f(x1, x2*2, c+1, pob)]
    if c % 2 != max(pob) % 2:
        return any(moves)
    else:
        return all(moves)

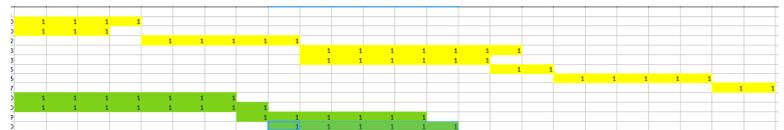
for s in range(1, 69 + 1):
    # if f(7, s, 0, [2]) == 1:
    #     print('№ 19:', s)
    if f(7, s, 0, [3]) == 1:
        print('№ 20:', s)
    if f(7, s, 0, [2, 4]) == 1 and f(7, s, 0, [2]) == 0:
        print('№ 21:', s)
```

Задача 22. Демо-2024

Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение четырёх процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

ID процесса	Время выполнения процесса	В (мс)	ID процессов	A
1	4	0		0
2	3	0		0
3	5	1:2		
4	7	3		3
5	6	3		3
6	2	5		5
7	5	4:6		6
8	2	7		7
9	7	0		0
10	8	0		0
11	6	9		9
12	6	10		10

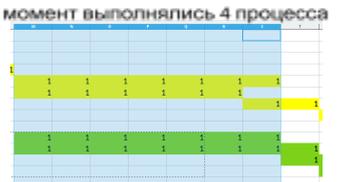
1) Заполняем шкалу и выделяем цветом взаимозависимые процессы.



2) Сдвигаем нижнюю шкалу так, чтобы в один момент выполнялись 4 процесса как можно дольше.

3) Считаем количество столбцов.

Ответ: 7



Задача 23. Демо-2024

Исполнитель преобразует число на экране.
У исполнителя есть три команды, которые обозначены латинскими буквами:

- A. Прибавить 1
- B. Умножить на 2
- C. Возвести в квадрат

Программа для исполнителя – это последовательность команд.
Сколько существует программ, для которых при исходном числе 2 результатом является число 20, при этом траектория вычислений не содержит числа 11?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы CBA при исходном числе 4 траектория будет состоять из чисел 16, 32, 33.

```
def f(x, y):  
    if x > y or x == 11: return 0  
    if x == y: return 1  
    return f(x + 1, y) + f(x * 2, y) + f(x ** 2, y)  
  
print(f(2, 20))
```

Задача 23. Демо-2025

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которые обозначены латинскими буквами:

- A. Вычти 2
- B. Найди целую часть от деления на 2

Программа для исполнителя – это последовательность команд.
Сколько существует программ, для которых при исходном числе 38 результатом является число 2 и при этом траектория вычислений содержит число 16?

```
def f(x, y):  
    if x < y: return 0  
    if x == y: return 1  
    return f(x - 2, y) + f(x // 2, y)  
  
print(f(38, 16) * f(16, 2))
```

Задача 24. Досрок

```
with open('24_7600.txt') as f:  
    s = f.readline().replace('Q', '!').replace('R', '!').replace('S', '!')  
    k = kmax = 1  
    for i in range(len(s) - 1):  
        if s[i] == s[i + 1] == '!':  
            k = i  
        else:  
            k += 1  
            kmax = max(kmax, k)  
print(kmax)
```

Текстовый файл состоит не более, чем из 1 200 000 прописных символов латинского алфавита. Определите максимальное количество идущих подряд символов, среди которых любые два символа из набора Q, R, S в различных комбинациях (с учётом повторений) не стоят рядом.

```
s=open('24_егел.txt').readline()  
k,l,lmax=0,0,0  
for r in range(2,len(s)):  
    if s[r-1]+s[r]=='CD':  
        k+=1  
    while k>160:  
        if s[l]+s[l+1]=='CD':  
            k-=1  
            l+=1  
        if k==160:  
            lmax=max(lmax,r-l+1)  
print(lmax)
```

Текстовый файл состоит из символов A, B, C, D, E и F. Определите максимальное количество идущих подряд символов в прилагаемом файле, среди которых пара символов CD (в указанном порядке) встречается ровно 160 раз. Для выполнения этого задания следует написать программу.

Задача 25. Досрок

```
for i in range(0, 10 ** 8 + 1, 273):  
    s = str(i)  
    if s[:2] == '12' and s[4:6] == '36' and s[-1] == '1':  
        print(i, i // 273)
```

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.

```
from fnmatch import *  
for x in range(0, 10 ** 10, 1917):  
    if fnmatch(str(x), '3?12?14*5'):  
        print(x, x // 1917)
```

Например, маске 123*4?5 соответствуют числа 123405 и 12300405. Среди натуральных чисел, не превышающих 10^8 , найдите все числа, соответствующие маске 12??36*1, делящиеся на 273 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 273.

Количество строк в таблице для ответа избыточно.

```
def f(n):  
    mass=[]  
    for i in range(2, round(n**0.5)+1):  
        if n%i == 0:  
            mass.append(i)  
            if i != n**0.5:  
                mass.append(n//i)  
    return mass
```

Задача 26. Досрок

```
with open('26_7602.txt') as f:  
    k = int(f.readline())  
    n = int(f.readline())
```

В аэропорту есть камера хранения из K ячеек, которые пронумерованы с 1. i принимаемый багаж кладется в свободную ячейку с минимальным номером. Известно время, когда пассажиры сдают и забирают багаж (в минутах с начала суток). Ячейка доступна для багажа, начиная со следующей минуты, после окончания срока хранения. Если свободных ячеек не находится, то багаж не принимается в камеру хранения.

Найдите количество багажа, которое будет сдано в камеры за 24 часа и номер ячейки, в которую сдаст багаж последний пассажир.

Входные данные

В первой строке входного файла находится число K – количество ячеек в камере хранения, во второй строке файла число N – количество пассажиров, сдающих багаж (натуральное число, не превышающее 1000). Каждая из следующих N строк содержит два натуральных числа, не превышающих 1440: время сдачи багажа и время выдачи багажа.

Выходные данные

Программа должна вывести два числа: количество сданных в камеру хранения багажа и номер ячейки, в которую примет багаж у последнего пассажира, который сможет сдать багаж.

Типовой пример организации данных:

```
2
5
30 60
40 60
50 1110
61 1010
1100 1440
```

Для указанного примера багаж смогут сдать первый, второй, четвёртый и пятый пассажир. Последний пассажир сдаст свой багаж в ячейку один, так как к этому моменту первая и вторая ячейка будут свободны.

Задача 27.

$$=СУММ(((A2-A$2:A$51)^2+(B2-B$2:B$51)^2)^{0,5})$$

-3,434404146	-3,749241428
-3,073661227	1,0200116

ctrl+shift+enter

СУММ(число1; [число2]; ...)

=ABS(СРЗНАЧ(H34:H35))*10000