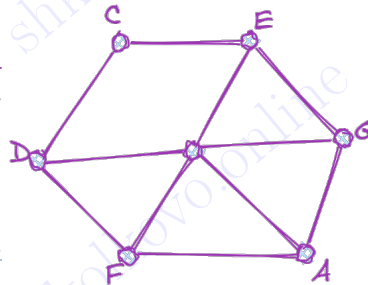


Информатика. ДВ. 19.06.2023

Задача 1

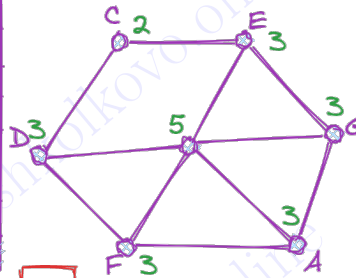
На рисунке схема дорог Н-ского района изображена в виде графа. Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какие номера в таблице соответствуют населённым пунктам F и G . В ответе запишите 2 числа в порядке возрастания.

	П1	П2	П3	П4	П5	П6	П7
П1		*	*	*	*		*
П2	*		*	*			
П3	*	*					*
П4	*	*			*		
П5	*			*		*	
П6					*	*	*
П7	*		*			*	



Решение

	П1	П2	П3	П4	П5	П6	П7
B		*	*	*	*		*
A	*		*	*			
F/G	*	*					*
F/G	*	*			*		
E/D	*			*		*	
C					*	*	*
E/D	*		*			*	



34

Задача 2

Алекс заполнил таблицу истинности логической функции F

$$(x \vee y) \wedge \overline{(y \equiv z)} \wedge \overline{w}$$

Он успел заполнить лишь фрагмент из трех различных её строк, даже не указав, какому столбцу таблицы соответствует каждая переменная x, y, z, w .

Определите, какому столбцу таблицы соответствует каждая из переменных x, y, z, w .

???	???	???	???	<i>F</i>
?	1	?	1	1
0	0	1	?	1
0	?	1	1	1

Решение

```
def f(x,y,z,w):
    return (x or y) and (not (y == z)) and (not w)

print("x y z w f")
for x in range(2):
    for y in range(2):
        for z in range(2):
            for w in range(2):
                if f(x,y,z,w) == 1:
                    print(x, y, z, w, f(x, y, z, w))
```

Проанализируем вывод нашей программы:

```
x y z w f
0 1 0 0 True
1 0 1 0 True
1 1 0 0 True
```

Первый столбик у нас это точно w , потому что все нули мы можем поставить только в первый. Далее мы можем сказать, что второй столбец — это z , ведь только там нет двух единиц, и может быть два нуля. Посмотрим на вторую строку, у нас стоит одна единица, два нуля и неизвестность. Только в данной строке мы можем сделать из нее три нуля и одну единицу, при этом данная единица будет принадлежать — y . Поэтому третий столбец — это y . И четвертый разумеется — это x .

Задача 3

Стандарт, вроде ВПР не требуется

Задача 4

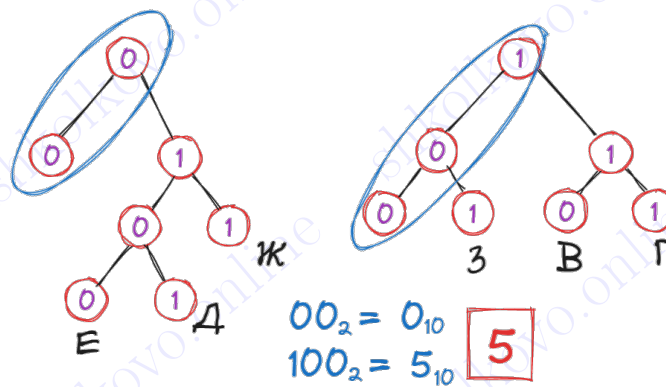
По каналу связи передаются сообщения, содержащие только восемь букв А, Б, В, Г, Д, Е, Ж и З. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны:

В	110
Г	111
Д	0101
Е	0100
Ж	011
З	101

Какое наименьшее количество двоичных знаков потребуется для кодирования оставшихся букв? В ответе запишите суммарную длину кодовых слов для букв: А, Б

Примечание: Условие Фано означает, что ни одно кодовое слово не является началом другого слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Решение



Задача 5

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

- 1) Строится троичная запись числа N

2) Если N не кратно 3, то остаток от деления на 3 умножается на 5, переводится в троичную запись и дописывается в конец числа

3) Результат R переводится в десятичную систему счисления и выводится на экран
Укажите минимальное число N , после обработки которого автомат получается число, большее 146.

Решение

```
def per(x): #ф-ция перевода в ЗСС
    s = ''
    while x > 0:
        s = str(x % 3) + s
        x //= 3
    return s

for n in range(1, 1000):
    x = per(n) #переводим в ЗСС
    if n % 3 == 0: #проверка исходного числа на делимость на 3
        x += x[-2:] #дописываем две последние цифры
        # в троичной записи
    else:
        x += str(per(n%3 * 5)) #дописываем в конец переведенный
        # в ЗСС остаток от деления на 3, умноженный на 5
    r = int(x, 3) #переводим результат в 10СС
    if r >= 146:
        print(n)
        break
```

Задача 6

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует три команды: **Вперёд** n (где n — целое число), вызывающая передвижение

Черепашки на n единиц в том направлении, куда указывает её голова, **Направо** m (где m — целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, и **Налево** m (где m — целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

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

Черепашке был дан для исполнения следующий алгоритм:

Повтори 2 [Вперёд 9 Направо 90 Вперёд 15 Направо 90]

Поднять хвост

Вперёд 12 Направо 90

Опустить хвост

Повтори 2 [Вперёд 6 Направо 90 Вперёд 12 Направо 90]

Определите, сколько точек с целочисленными координатами будут находиться внутри пересечения фигур, ограниченных заданными алгоритмом линиями, включая точки на границах этого пересечения.

Решение в Кумире:

использовать Черепашка

алг

нач

опустить хвост

нц 2 раз

вперед(9)

вправо(90)

вперед(15)

вправо(90)

кц

поднять хвост

вперед(12)

вправо(90)

опустить хвост

нц 2 раз

вперед(6)

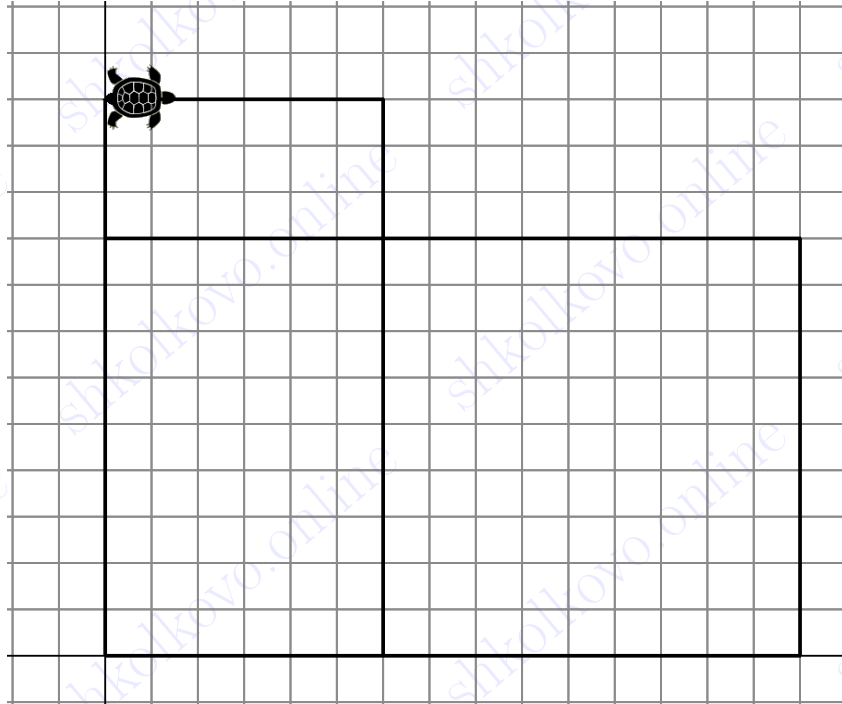
вправо(90)

вперед(12)

вправо(90)

кц

кон



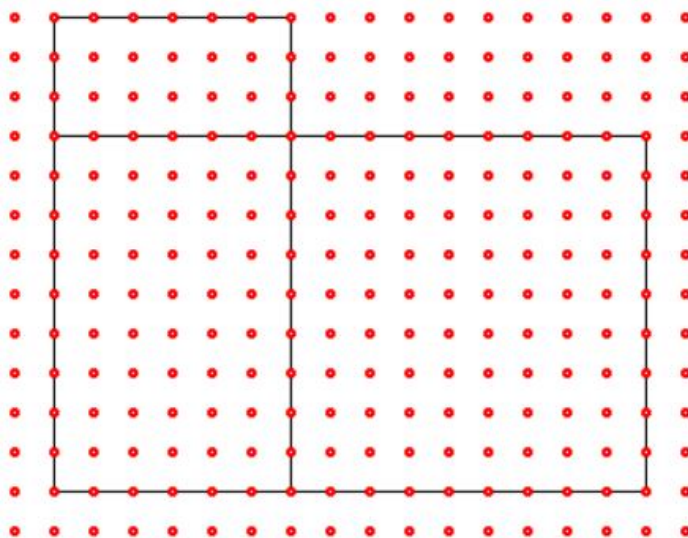
Решение в Питоне:

```
from turtle import *
left(90)
k = 20
tracer(3)
for i in range(2):
    forward(9 * k)
    right(90)
    forward(15 * k)
    right(90)
pu()
forward(12 * k)
right(90)
pd()
for i in range(2):
```

```

forward(6*k)
right(90)
forward(12 * k )
right(90)
pu()
for x in range(-20, 20):
    for y in range(-20, 20):
        goto(x * k, y * k)
        dot(5, "red")
        dot(2, "white")
mainloop()

```



Задача 7

Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 1280 × 720 пикселей, при условии, что цвет каждого пикселя кодируется 4 Байтами.

Решение

$$\frac{I}{V} = \frac{k \cdot i}{V} = \frac{1280 \cdot 720 \cdot 32}{28800} = 1024 \text{ секунд.}$$

Задача 8

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сооб-

щению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует пятибуквенные слова, в которых могут быть только буквы ПЯТНИЦА, причём буква Я появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. На первом месте НЕ может стоять буква Н. Сколько различных кодовых слов может использовать Игорь?

Решение через itertools

```
from itertools import product
count = 0
for word in product('ПЯТНИЦА', repeat=5):
    word = "".join(word)
    if word[0] != "Н" and word.count("Я") == 1:
        count += 1
print(count)
```

Решение без itertools

```
count = 0
for a in 'ПЯТНИЦА':
    for b in 'ПЯТНИЦА':
        for c in 'ПЯТНИЦА':
            for d in 'ПЯТНИЦА':
                for e in 'ПЯТНИЦА':
                    word = a + b + c + d + e
                    if word[0] != "Н" and word.count("Я") == 1:
                        count += 1
print(count)
```

Задача 9

В файле находится таблица, которая содержит в каждой из строк по 7 натуральных чисел. Найдите количество таких строк, в которых два числа повторяются по 2 раза, а 3 других различны, среднее арифметическое неповторяющихся чисел больше среднего арифметического повторяющихся.

Задача 10

С помощью текстового редактора определите, сколько раз, не считая сносок, встречается слово «солдат» (со строчной буквы) в тексте повести Александра Куприна «Поединок». Однокоренные слова не следует учитывать. В ответе укажите только число.

Задача 11

При регистрации в компьютерной системе каждому объекту присваивается идентификатор, состоящий из 65 символов и содержащий только десятичные цифры и символы из 2500-символьного специального алфавита. В базе данных для хранения сведений о каждом идентификаторе отведено одинаковое и минимальное 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 сведений о 16384 объектах.

В ответе запишите только целое число - количество Кбайт

Решение

Алфавит: 2500 символов, кодируем каждый символ 12 битами.

Один идентификатор кодируется $\frac{12 \cdot 65}{8} = 97.5$, но так как одинаковое и минимальное целое число БАЙТ, то округляем до 98

$$\frac{98 \cdot 16384}{1024} = 1568$$

Задача 12

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в которых v и w обозначают последовательности цифр:

А) заменить (v, w) .

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w .

Например, выполнение команды заменить(111, 27) преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений последовательности v , то выполнение команды заменить (v, w) не меняет эту строку.

Б) Нашлось (v) .

Эта команда проверяет, встречается ли последовательность v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь».

Строка исполнителя при этом не изменяется.

ПОКА нашлось (52) ИЛИ нашлось (1122) ИЛИ нашлось (2222)

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» ($3 < n < 10000$). Определите наименьшее значение n , при котором сумма цифр в строке, получившаяся в результате выполнения программы, равна 64.

Решение

```
for n in range(4,10000):
    s = '5' + '2' * n
    while ('52' in s) or ('1122' in s) or ('2222' in s):
        if ('52' in s):
            s = s.replace('52', '11', 1)
        if ('2222' in s):
            s = s.replace('2222', '5', 1)
        if ('1122' in s):
```

```

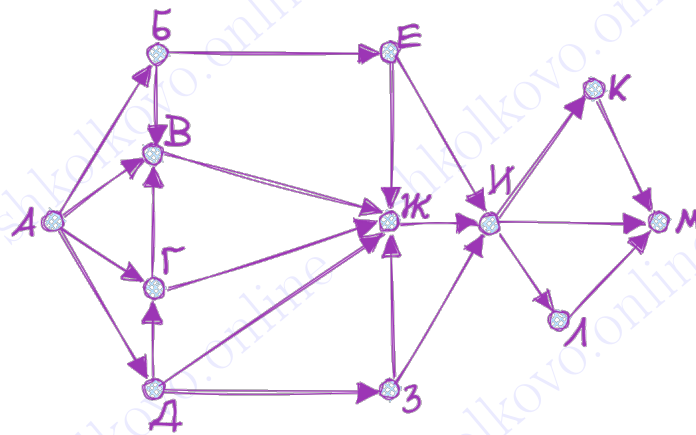
s = s.replace('1122', '25', 1)
r = s.count('1') * 1 + s.count('2') * 2 + s.count('5') * 5
if r == 64:
    print(n)
    break

```

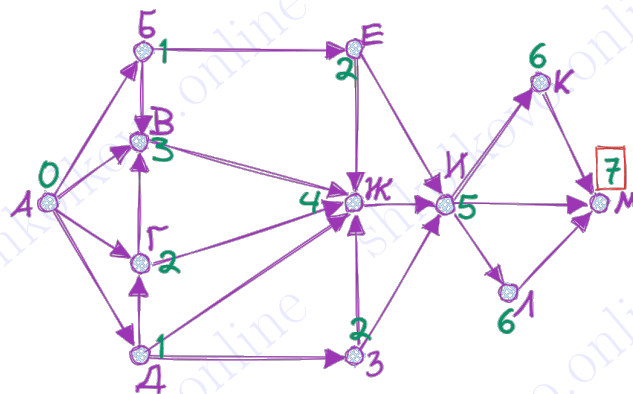
Ответ: 152

Задача 13

На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Какова длина самого длинного пути из города А в город М? Длиной пути считать количество дорог, составляющих этот путь.



Решение



Задача 14

Операнды арифметического выражения записаны в системе счисления с основанием 22

$$98x79641_{22} + 25x43_{22} + 63x5_{22}$$

В записи чисел переменной x обозначена неизвестная цифра алфавита 15-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения кратно 21. Для найденного значения x вычислите частно от деления значения арифметического выражения на 21 и укажите его в ответе в десятичной системе счисления.

Основание системы счисления указывать не нужно.

Решение

```
for x in '0123456789ABCDE':
    if (int('98' + x + '79641', 22) + int('25' + x + '43', 22) +
        int('63' + x + '5', 22)) % 21 == 0:
        print((int('98' + x + '79641', 22) + int('25' + x + '43', 22) +
            int('63' + x + '5', 22)) // 21)
        break
```

Задача 15

Для какого наибольшего целого неотрицательного числа A выражение

$$(y + 3x > A) \vee (x < 20) \vee (y < 50)$$

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

Решение

```
def f(x, y, A):
    return (y + 3*x > A) or (x < 20) or (y < 50)

for a in range(1000):
    if all(f(x, y, a) == 1 for x in range(100) for y in range(100)):
```

```
print(a)
```

Задача 16

Задан алгоритм вычисления функции $F(n)$, где n - натуральное число:

$F(n) = 1$, при $n = 1$;

$F(n) = n - 2 + F(n - 1)$, если $n > 1$

Чему равно значение функции $F(2023) - F(2021)$?

Решение

```
from functools import lru_cache
```

```
@lru_cache(None)
```

```
def f(n):
```

```
    if n == 1:
```

```
        return 1
```

```
    elif n > 1:
```

```
        return n - 2 + f(n-1)
```

```
for i in range(1, 2023):
```

```
    f(i)
```

```
print(f(2023) - f(2021))
```

Задача 17

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -1000000 до 1000000 включительно. Определите количество троек элементов в которых только одно число трехзначное, и сумма элементов тройки меньше максимального числа последовательности оканчивающегося на 17. В ответе запишите два числа: сначала количество найденных троек, а затем минимальную из сумм таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Решение

```

f = open('17.txt')
a = list(map(int, f.readlines()))
m = max([c for c in a if str(c)[-2:] == '17'])
n = len(a)
k = 3
count = 0
ans = 10 ** 20
for i in range(n - (k - 1)):
    x = a[i : i + k]
    l = len([c for c in x if 99 < abs(c) < 1000])
    if l == 1 and sum(x) < m:
        count += 1
        ans = min(sum(x), ans)
print(count, ans)

```

Задача 18

Квадрат разлинован на $N \times N$ клеток ($1 < N < 17$). 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. При попытке выхода за границу квадрата Робот разрушается. Между соседними клетками квадрата 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 клетке маршрута Робота.

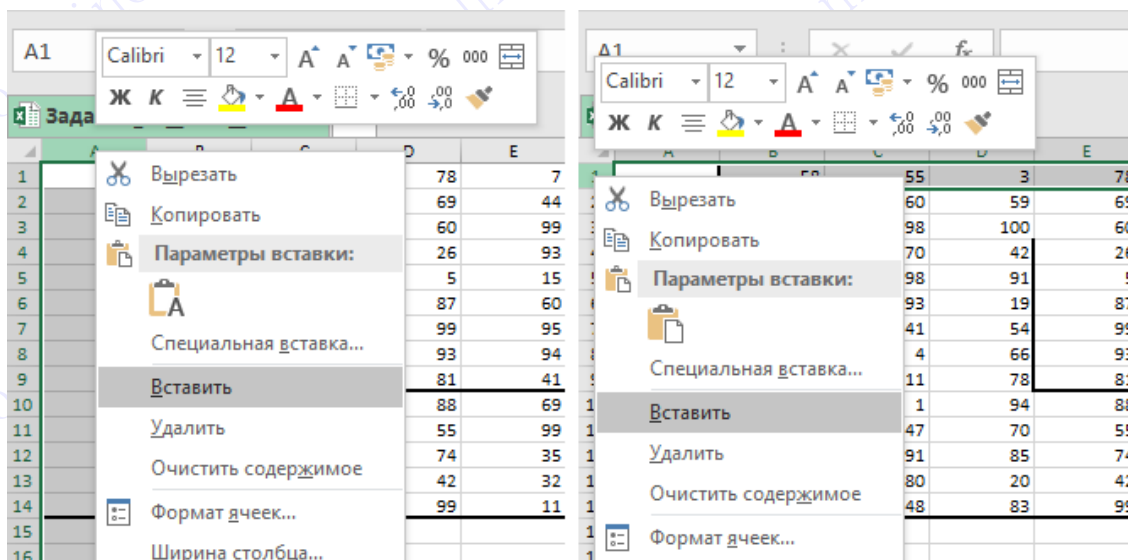
Откройте файл. Определите максимальную денежную сумму, которую может собрать Робот, начиная в верхнем левом углу. В ответ запишите одно число — максимальную сумму, которую может собрать Робот.

Исходные данные представляют собой электронную таблицу размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Решение

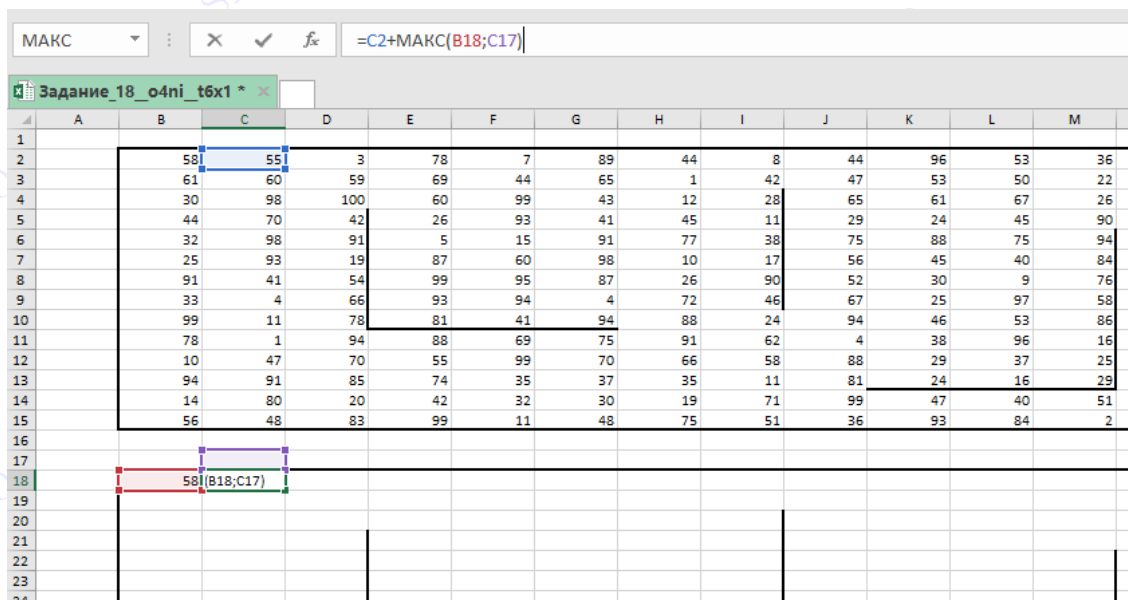
Добавляем пустой столбец (нажимаем правой кнопкой мыши на столбец A и вы-

бираем *Вставить*) и пустую строку перед самой первой (нажимаем правой кнопкой мыши на строку 1 и выбираем *Вставить*).



Копируем таблицу и с помощью специальной вставки ($Ctrl + Alt + V$) вставляем только её формат.

В начало маршрута (в нашем случае $B18$) записываем значение левой верхней клетки данной нам таблицы. В клетку $C18$ записываем формулу $=C2+МАКС(B18;C17)$



Раполним ей все оставшиеся ячейки таблицы. Теперь изменим формулы там, где у нас находятся границы. Если границы мешают нам пройти ниже, то меняем формулы в клетках, находящихся под границами, то есть убираем из них значение ячейки идущей выше границы.

	B	C	D	E	F	G	H	I	J	K	L	M
8	91	41	54	99	95	87	26	90	52	30	9	
9	33	4	66	93	94	4	72	46	67	25	97	
10	99	11	78	81	41	94	88	24	94	46	53	
11	78	1	94	88	69	75	91	62	4	38	96	
12	10	47	70	55	99	70	66	58	88	29	37	
13	94	91	85	74	35	37	35	11	81	24	16	
14	14	80	20	42	32	30	19	71	99	47	40	
15	56	48	83	99	11	48	75	51	36	93	84	
16												
17												
18	58	113	116	194	201	290	334	342	386	482	535	
19	119	179	238	307	351	416	417	459	506	559	609	
20	149	277	377	437	536	579	591	619	571	632	699	
21	193	347	419	=E5+E20	629	670	715	726	600	656	744	
22	225	445	536	468	644	761	838	876	675	763	838	
23	250	538	557	555	704	859	869	893	731	808	878	
24	341	579	633	654	799	946	972	1062	783	838	887	
25	374	583	699	747	893	950	1044	1108	850	875	984	
26	473	594	777	828	934	1044	1132	1156	1250	1296	1349	
27	551	595	871	959	1028	1119	1223	1285	1289	1334	1445	
28	561	642	941	1014	1127	1197	1289	1347	1435	1464	1501	
29	655	746	1026	1100	1162	1234	1324	1358	1516	1540	1556	
30	669	826	1046	1142	1194	1264	1343	1429	1615	1662	1702	

Если граница мешает нам идти вправо, то убираем из текущей формулы значение ячейки, идущей после границы.

	E	F	G	H	I	J	K	L	M	N	O	P
8	99	95	87	26	90	52	30	9	76	86	18	
9	93	94	4	72	46	67	25	97	58	58	10	
10	81	41	94	88	24	94	46	53	86	11	28	
11	88	69	75	91	62	4	38	96	16	90	24	
12	55	99	70	66	58	88	29	37	25	18	83	
13	74	35	37	35	11	81	24	16	29	4	88	
14	42	32	30	19	71	99	47	40	51	98	68	
15	99	11	48	75	51	36	93	84	2	14	80	
16												
17												
18	194	201	290	334	342	386	482	535	571	656	740	
19	307	351	416	417	459	506	559	609	631	699	801	
20	437	536	579	591	619	571	632	699	725	731	826	
21	463	629	670	715	726	600	656	744	834	860	953	
22	468	644	761	838	876	675	763	838	932	882	971	
23	555	704	859	869	893	731	808	878	1016	981	989	
24	654	799	946	972	1062	783	838	887	1092	1067	1085	
25	747	893	950	1044	1108	850	875	984	1150	1125	1135	
26	828	934	1044	1132	1156	1250	1296	1349	1435	1136	1164	
27	959	1028	1119	1223	1285	1289	1334	1445	1461	1226	1250	
28	1014	1127	1197	1289	1347	1435	1464	1501	1526	1244	1333	
29	1100	1162	1234	1324	1358	1516	1540	1556	1585	1248	1421	
30	1142	1194	1264	1343	1429	1615	=K14+J30	1702	1753	1851	1919	
31	1241	1252	1312	1418	1480	1651	1755	1839	1841	1865	1999	
32												
33												

Выписываем значение из ячейки O31 в ответ.

Задача 19

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит ку-

ча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один или пять камней, либо увеличить количество камней в куче в четыре раза. У каждого игрока есть неограниченное количество камней, чтобы делать ходы. Игра завершается в тот момент, когда количество камней в куче становится не менее 473.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу из 473 камней или больше.

В начальный момент в куче было S камней: $1 \leq S \leq 472$.

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.

Решение

```
import sys
sys.setrecursionlimit(100000)

from functools import lru_cache

def moves(h):
    return h + 1, h + 5, h * 4

@lru_cache(None)
def f(h):
    if h >= 473:
        return 'END'
    if any(f(x) == 'END' for x in moves(h)):
        return 'P1'
    if all(f(x) == 'P1' for x in moves(h)):
        return 'V1'
    if any(f(x) == 'V1' for x in moves(h)):
        return 'P2'
    if all(f(x) == 'P1' or f(x) == 'P2' for x in moves(h)):
        return 'V2'
```

```
for s in range(1, 472):
    if f(s) == 'V1':
        print(s)
```

Задача 20

Для игры, описанной в предыдущем задании, найдите два таких значения S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть за один ход и Ваня может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

В ответе запишите числа в порядке возрастания без пробелов и знаков препинаний.

Решение

```
import sys
sys.setrecursionlimit(100000)

from functools import lru_cache

def moves(h):
    return h + 1, h + 5, h * 4

@lru_cache(None)
def f(h):
    if (h >= 473):
        return 'END'
    if any(f(x) == 'END' for x in moves(h)):
        return 'P1'
    if all(f(x) == 'P1' for x in moves(h)):
        return 'V1'
    if any(f(x) == 'V1' for x in moves(h)):
        return 'P2'
    if all(f(x) == 'P1' or f(x) == 'P2' for x in moves(h)):
        return 'V2'
```

```
for s in range(1, 472):
    if f(s) == 'P2':
        print(s)
```

Задача 21

Для игры, описанной ранее, найдите два таких 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.

В ответе запишите числа в порядке возрастания без пробелов и знаков препинаний.

Решение

```
import sys
sys.setrecursionlimit(100000)

from functools import lru_cache

def moves(h):
    return h + 1, h + 5, h * 4

@lru_cache(None)
def f(h):
    if (h >= 473):
        return 'END'
    if any(f(x) == 'END' for x in moves(h)):
        return 'P1'
    if all(f(x) == 'P1' for x in moves(h)):
        return 'V1'
    if any(f(x) == 'V1' for x in moves(h)):
        return 'P2'
```

```

if all(f(x) == 'P1' or f(x) == 'P2' for x in moves(h)):
    return 'V2'

for s in range(1, 472):
    if f(s) == 'V2':
        print(s)

```

Задача 22

В файле содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс B зависит от процесса A , если для выполнения процесса B необходимы результаты выполнения процесса A . В этом случае процессы могут выполняться только последовательно.

Информация о процессах представлена в файле в виде таблицы. В первой строке таблицы указан идентификатор процесса (ID), во второй строке таблицы — время его выполнения в миллисекундах, в третьей строке перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Типовой пример организации данных в файле:

ID процесса B	Время выполнения процесса B (мс)	ID процесса(ов) A
1	4	0
2	3	0
3	1	1; 2
4	7	3

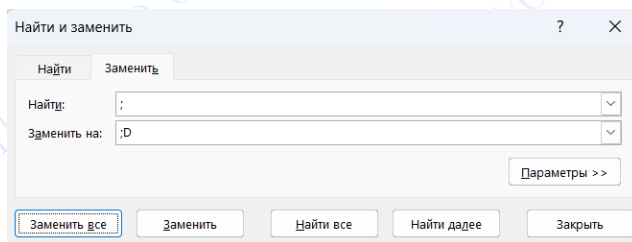
Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Решение Excel (пример)

Конечно, можно вручную посчитать время для каждого процесса, но это займёт слишком много времени, к тому же, вдруг файл будет состоять из миллиона строк?

Как считать тогда? Поэтому приведём более быстрое решение.

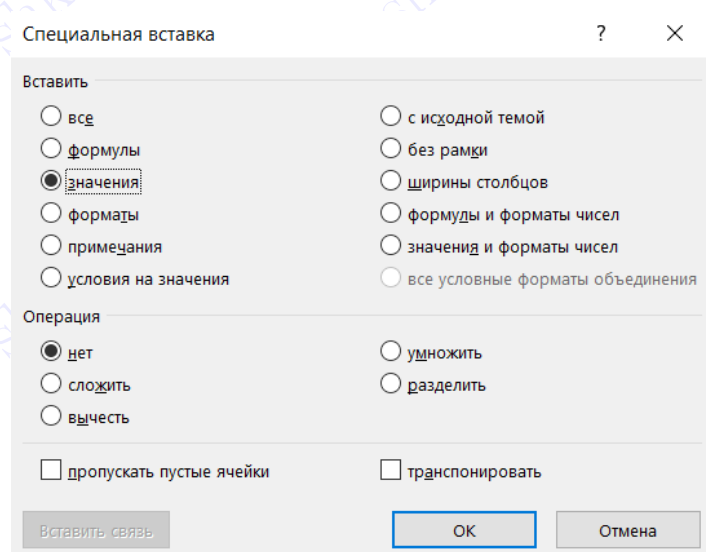
Удаляем первую строку с заголовком. С помощью комбинации клавиш Ctrl+N заменяем все «;» на «;D», как показано ниже.



В ячейку D1 записываем формулу =ЕСЛИ(C1=0;B1;"МАКС(D"&C1&"")+&B1) и растягиваем на всю таблицу, как показано ниже.

D1	A	B	C	D	E	F
1	1	91	0	91		
2	2	758	0	758		
3	3	748	0	748		
4	4	28	0	28		
5	5	667	1;D3	МАКС(D1;D3)+667		
6	6	729	0	729		
7	7	884	2;D5	МАКС(D2;D5)+884		
8	8	631	2	МАКС(D2)+631		
9	9	794	3	МАКС(D3)+794		
10	10	850	2	МАКС(D2)+850		
11	11	455	0	455		
12	12	309	4;D10	МАКС(D4;D10)+309		
13	13	529	1	МАКС(D1)+529		
14	14	124	10;D12	МАКС(D10;D12)+124		
15	15	522	0	522		
16	16	699	14	МАКС(D14)+699		
17	17	796	14;D15	МАКС(D14;D15)+796		
18	18	749	0	749		
19	19	76	0	76		
20	20	969	5;D10	МАКС(D5;D10)+969		
21	21	36	8;D11	МАКС(D8;D11)+36		
22	22	532	6	МАКС(D6)+532		
23	23	108	0	108		
24	24	395	19	МАКС(D19)+395		
25	25	820	5	МАКС(D5)+820		

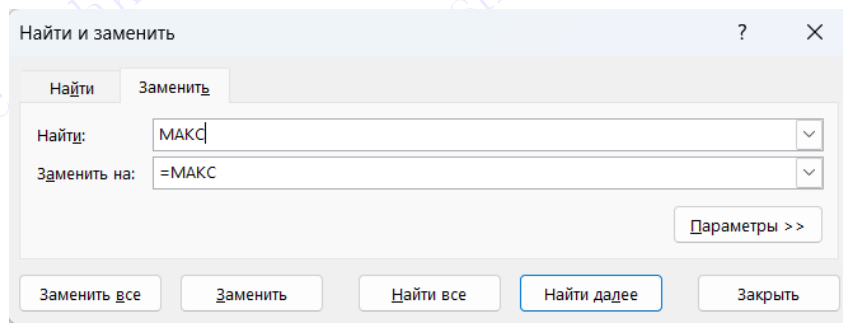
Копируем столбец D, и в ячейку E1 вставляем только значения из этого столбца, как показано ниже.



Теперь очищаем столбец *D*, копируем значения из столбца *E* и вставляем в столбец *D*. Столбец *E* нам больше не понадобится, можем его удалить.

	A	B	C	D	E
1	1	91	0	91	
2	2	758	0	758	
3	3	748	0	748	
4	4	28	0	28	
5	5	667	1;D3	МАКС(D1;D3)+667	
6	6	729	0	729	
7	7	884	2;D5	МАКС(D2;D5)+884	
8	8	631	2	МАКС(D2)+631	
9	9	794	3	МАКС(D3)+794	
10	10	850	2	МАКС(D2)+850	
11	11	455	0	455	
12	12	309	4;D10	МАКС(D4;D10)+309	
13	13	529	1	МАКС(D1)+529	
14	14	124	10;D12	МАКС(D10;D12)+124	
15	15	522	0	522	
16	16	699	14	МАКС(D14)+699	
17	17	796	14;D15	МАКС(D14;D15)+796	
18	18	749	0	749	
19	19	76	0	76	
20	20	969	5;D10	МАКС(D5;D10)+969	
21	21	36	8;D11	МАКС(D8;D11)+36	
22	22	532	6	МАКС(D6)+532	
23	23	108	0	108	
24	24	395	19	МАКС(D19)+395	
25	25	820	5	МАКС(D5)+820	

И последнее: с помощью комбинации клавиш *Ctrl* + *H* заменяем все «МАКС» на «=МАКС», как показано ниже.



В ячейку *E1* записываем формулу «=МАКС(D:D)» и получаем ответ.

	A	B	C	D	E	F
1	1	91	0	91	5565	
2	2	758	0	758		
3	3	748	0	748		
4	4	28	0	28		
5	5	667	1,03	1415		
6	6	729	0	729		
7	7	884	2,05	2299		
8	8	631	2	1389		
9	9	794	3	1542		
10	10	850	2	1608		
11	11	455	0	455		
12	12	309	4,010	1917		
13	13	529	1	620		
14	14	124	10,012	2041		
15	15	522	0	522		
16	16	699	14	2740		
17	17	796	14,015	2837		
18	18	749	0	749		
19	19	76	0	76		
20	20	969	5,010	2577		
21	21	36	8,011	1425		
22	22	532	6	1261		
23	23	108	0	108		
24	24	395	19	471		
25	25	820	5	2235		

Решение прогой (пример)

Для нашего удобства удалим первую строку таблицы. Далее предлагаю изменить нумерацию ID процессов.

```
n=100
```

```
a=[input() for i in range(n)]
```

```
for i in range(n):
```

```
    s=a[i].split()
```

```
    id_old=s[0]
```

```

    id_new=str(i+1)
    for j in range(n):
        a[j]=a[j].replace(id_old,id_new)

for i in range(n):
    print(a[i])

```

Преобразуем данные третьего столбца в готовые формулы для Excel таблицы.

```

n=100
for i in range(1,n+1):
    s=input().split(';')
    if s[0]=='0':
        formula = '=B' + str(i)
    else:
        formula = '=МАКС('
        for x in s:
            formula = formula+'D'+x+';'
        formula=formula[:-1] +'') + B'+ str(i)
    print(formula)

```

С помощью специальной вставки (Ctrl+Alt+V) вставляем готовые формулы в таблицу.

В ячейку F1 записываем формулу «=МАКС(D1:D100)» и получаем ответ.

Задача 23

Исполнитель преобразует число, записанное на экране.

У исполнителя есть команды, которым присвоены номера:

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

Первая команда увеличивает число на экране на 1, вторая — на 3, третья — увеличивает число в 3 раза.

Сколько существует программ, для которых при исходном числе 7 результатом является число 20 и при этом траектория содержит число 14 и не содержит 15?

Динамическое решение:

```
a = [0]*1000
a[7] = 1
for i in range(8, 20 + 1):
    a[i] = a[i - 1] + a[i - 3]
    if i % 3 == 0:
        a[i] += a[i // 3]
    if i == 15:
        a[i] = 0
    if i == 14:
        for j in range(14):
            a[j] = 0
print(a[20])
```

Рекурсивное решение:

```
def f(start, stop):
    if start > stop or start == 15:
        return 0
    if start == stop:
        return 1
    return f(start + 1, stop) + f(start + 3, stop) + f(start * 3, stop)
print(f(7, 14) * f(14, 20))
```

Задача 24

Текстовый файл состоит из символов, обозначающих прописные буквы латинского алфавита. Определите максимальное количество идущих подряд символов, которых никакие две буквы из набора букв *A*, *B* и *C* (с учетом повторений) не записаны подряд.

Для выполнения этого задания следует написать программу.

Решение

```
f = open()
```

```

s = f.readline()
count = 1
maxim = 1
a = ['AA', 'AB', 'AC', 'BA', 'BB', 'BC', 'CA', 'CB', 'CC']
for i in range(len(s) - 1):
    if (s[i] + s[i + 1]) not in a:
        count += 1
        maxim = max(maxim, count)
    else:
        count = 1
print(maxim)

```

Задача 25

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 .

Найдите все натуральные числа, не превосходящие 10^9 , для которых выполнены все условия:

- соответствуют маске $*31 * 65?$;
- делятся на 31 и 2031 без остатка;
- количество делителей числа является результатом любой степени двойки.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, справа от каждого числа их частное от деления на 2031 .

Решение №1:

```

def countDivs(n):
    a = set()
    for i in range(1, int(n ** 0.5) + 1):
        if n % i == 0:
            a.add(i)

```

```

        a.add(n // i)
    return len(a)

def degreeOfTwo(n):
    return n & (n - 1) == 0

for num in range(31 * 2031, 10**9 + 1, 31 * 2031):
    s = str(num)
    flag = degreeOfTwo(countDivs(num))
    if flag and s.count("31") != 0 and s[-3:-1] == "65":
        print(num, num // 2031)

```

Решение №2:

```

def count_div(n):
    k = 0
    for j in range(1, int(n **0.5) + 1):
        if n % j == 0:
            k += 1
            if j != n // j:
                k += 1
    return k

def dvoika(n):
    m = n
    while m > 1:
        if m % 2 == 0:
            m //= 2
        else:
            return False
    return True

start = 0
for i in range(31650, 10 ** 9):

```

```

if i % 31 == 0 and i % 2031 == 0 and str(i)[-3:-1] == '65' and \
    str(i).count('31') > 0:
    start = i
    break

```

```

for i in range(start, 10**9 + 1, 31 * 2031):
    counter = count_div(i)
    if dvoika(counter) and str(i)[-3:-1] == '65' and \
        str(i).count('31') > 0:
        print(i, i // 2031)

```

Решение №3:

```

from fnmatch import *

```

```

def countDivs(n):
    a = set()
    for i in range(1, int(n ** 0.5) + 1):
        if n % i == 0:
            a.add(i)
            a.add(n // i)
    return len(a)

```

```

for x in range(31 * 2031, 10**9 + 1, 31 * 2031):
    n = countDivs(x)
    if fnmatch(str(x), "*31*65?") and n & (n - 1) == 0:
        print(x, x // 2031)

```

Задача 26

Задание 26. Дальний восток

Входной файл содержит информацию о плане проведения собраний в конференц-зале. Для каждого собрания известно время проведения и длительность собрания. Определите максимальное количество собраний, которое будет проведено и в какую

минуту завершиться последнее собрание. Если способов выбрать последнее собрание несколько, выбрать нужно то, длительность которого больше.

В первой строке входного файла находится натуральное число N , ($N \leq 1000$), обозначающее количество собраний. Каждая из следующих N строк содержит два натуральных числа: указанное в заявке время проведения (в минутах от начала суток, не превышает 1300) и длительность (в минутах, не превышает 1000) собрания.

Запишите в ответ два числа: максимальное количество собраний, которое будет проведено и в какую минуту завершиться последнее собрание. Типовой пример организации данных во входном файле

```
5
10 150
100 110
120 130
131 150
131 180
```

Ответ к примеру: 3 180

Похоже на задачу про юникс-время, но найти надо другое: не момент когда процессов много одновременно, а что-то (ждём точную формулировку), предположительно макс. количество конференций которые удастся провести, тогда жадник + сканлайн

Разбирали на этих вебах <https://2.shkolkovo.online/courses/5/1294/11171>,
<https://2.shkolkovo.online/courses/5/1455/11180>

Задача 27

В первой строке подаются два натуральных числа N — количество натуральных чисел в последовательности и K — минимальное расстояние, допустимое между любыми двумя элементами.

Требуется найти минимальное значение произведения тройки элементов так, что между любыми элементами тройки расстояние между двумя элементами не менее K (т.е. разность их индексов по модулю больше или равна K).

Решение

```
from random import randint
```



```

def neeff(a, k):
    n = len(a)
    pr = 10000000000000000
    for i in range(n):
        for j in range(i+k, n):
            for p in range(j+k, n):
                pr = min(pr, a[i]*a[j]*a[p])
    return pr

def eff(a, k):
    n = len(a)
    number = 10**15
    pair = 10**15
    min_pr = 10**15

    for i in range(2*k, n):
        # Мин. допустимое расстояние от самого правого
        # числа тройки до самого левого
        number = min(number, a[i - 2*k])

        # Мин. допустимое расстояние от второго
        # числа тройки до самого левого
        # (обычная пара с расстоянием)
        pair = min(pair, number * a[i - k])

        # Составили допустимую пару с расстоянием
        # K или более. Теперь считаем произведение пары
        # на третий элемент, т.е. считаем тройку
        min_pr = min(min_pr, pair * a[i])

    return min_pr

```

```
for j in range(100):
    n = 200
    k = randint(2, 30)
    a = [randint(2, 10000) for i in range(n)]
    if eff(a, k) != neeff(a, k):
        print('ОШИБКА')
        break
```