

## Содержание

<b>Задача 1</b>	<b>5</b>
№1.1 . . . . .	5
№1.2 (от 8.06) . . . . .	6
<b>Задача 2</b>	<b>7</b>
№2.1 . . . . .	7
№2.2 (от 8.06) . . . . .	8
<b>Задача 3</b>	<b>9</b>
Подсказка . . . . .	9
<b>Задача 4</b>	<b>10</b>
№4.1 . . . . .	10
<b>Задача 5</b>	<b>11</b>
№5.1 . . . . .	11
№5.2 (от 8.06) . . . . .	12
<b>Задача 6</b>	<b>13</b>
№6.1 . . . . .	13
№6.2 (от 8.06) . . . . .	15
<b>Задача 7</b>	<b>18</b>
№7.1 . . . . .	18
№7.2 (от 8.06) . . . . .	18
<b>Задача 8</b>	<b>19</b>
№8.1 . . . . .	19
№8.2 (от 8.06) . . . . .	19

№8.3 (от 8.06)	21
<b>Задача 9</b>	<b>22</b>
№9.1	22
№9.2 (от 8.06)	22
№9.3 (от 8.06)	23
<b>Задача 10</b>	<b>24</b>
№10.1	24
<b>Задача 11</b>	<b>24</b>
№11.1	24
№11.2 (от 8.06)	25
№11.3 (от 8.06)	25
№11.4 (от 8.06)	26
<b>Задача 12</b>	<b>26</b>
№12.1	26
№12.2 (от 8.06)	27
<b>Задача 13</b>	<b>28</b>
№13.1	28
№13.2 (от 8.06)	29
№13.3 (от 8.06)	30
№13.4 (от 8.06)	31
<b>Задача 14</b>	<b>32</b>
№14.1	32
№14.2 (от 8.06)	33
<b>Задача 15</b>	<b>33</b>
№15.1	33
№15.2 (от 8.06)	34
№15.3 (от 8.06)	35

<b>Задача 16</b>	<b>35</b>
№16.1	35
№16.2 (от 8.06)	36
<b>Задача 17</b>	<b>37</b>
№17.1	37
№17.2 (от 8.06)	38
<b>Задача 18</b>	<b>39</b>
№18.1	39
<b>Задача 19-21</b>	<b>41</b>
№19-21.1	41
№19-21.2 (от 8.06)	43
<b>Задача 22</b>	<b>45</b>
№22.1	45
<b>Задача 23</b>	<b>46</b>
№23.1	46
№23.2 (от 8.06)	47
<b>Задача 24</b>	<b>48</b>
№24.1	48
№24.2 (от 8.06)	48
<b>Задача 25</b>	<b>50</b>
№25.1	50
№25.2 (от 8.06)	51
<b>Задача 26</b>	<b>52</b>
№26.1	52
№26.2 (от 8.06)	56
№26.3 (от 8.06)	57

**Задача 27**

**59**

№27.1 ..... 59

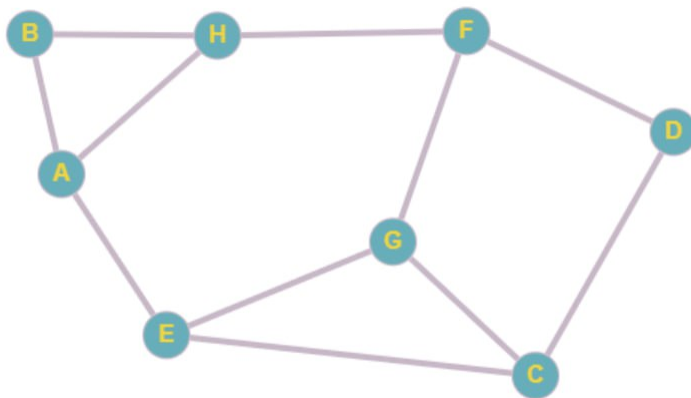
№27.2 (от 8.06) ..... 61

## Задача 1

### №1.1

На рисунке схема дорог N-ского района изображена в виде графа, в таблице содержатся сведения о протяженности каждой из этих дорог (в километрах).

	1	2	3	4	5	6	7	8
1		53		1			2	
2	53			13				8
3					30		39	74
4	1	13				5		
5			30					3
6				5			21	
7	2		39			21		
8		8	74		3			



Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённостей дорог из пункта В в пункт Н и из пункта А в пункт Е.

В ответе запишите целое число.

### Решение

Е и G – уникальные пункты, так как это троичные пункты, которые связаны с тремя троичными пунктами, таким образом С определяется уникально, так как

связано сразу с Е и G.

Е и G это 1 и 2, тогда С это 4.

Третья связь С – это D, значит D – 6.

D связан с F, F – это 7.

F связан с H и G, G это 1 или 2, значит H это 3, а G – это 1, Е – 2.

H связан с двоичным В, значит В это 5.

А это 8.

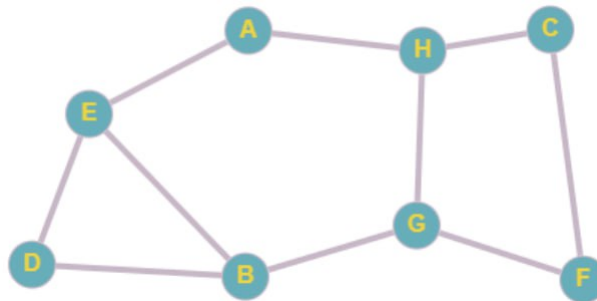
Значит искомые суммы это 30 и 8.  $30 + 8 = 38$

**Ответ:** 38

### №1.2 (от 8.06)

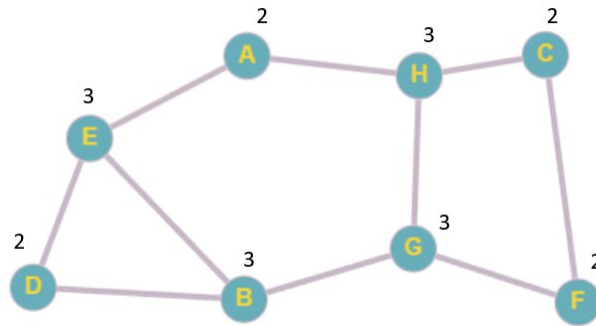
На рисунке схема дорог N-ского района изображена в виде графа, в таблице содержатся сведения о протяженности каждой из этих дорог (в километрах).

	1	2	3	4	5	6	7	8
1			2					6
2					12			31
3	2			35		3		
4			35			5		
5		12					17	
6			3	5			1	
7					17	1		93
8	6	31					93	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённостей дорог из пункта А в пункт Е и из пункта А в пункт Н. В ответе запишите целое число.

## Решение



Подпишем около каждой вершины ее степень (количество ребер, выходящих из нее). Получается:  $G - 3$ ,  $C - 2$ ,  $D - 2$ ,  $C - 3$ ,  $F - 2$ ,  $E - 3$ ,  $A - 2$ ,  $H - 3$ .

Вершина  $A$  единственная вершина со степенью 2, которая связана с двумя вершинами степени 3, которые не связаны между собой, тогда  $A - П1$ . Вершина  $D$  единственная вершина со степенью 2, которая связана с двумя вершинами степени 3, которые связаны между собой, тогда  $D - П4$ . Вершина  $E$  связана и с  $D$  и с  $A$ , поэтому  $E - П3$ , тогда  $B - П6$ . Найти остальные вершины не составит труда:  $G - П7$ ,  $H - П8$ ,  $C - П2$ ,  $F - П5$ .

Протяжённость дорог из пункта  $A$  ( $П1$ ) в  $E$  ( $П3$ ) равна 2, а из  $A$  ( $П1$ ) в  $H$  ( $П8$ ): 6.

Найдём их сумму:  $2 + 6 = 8$

**Ответ:** 8

## Задача 2

### №2.1

Миша заполнял таблицу истинности логической функции  $F$

$$((x \rightarrow y) \rightarrow z) \vee \bar{w}$$

Но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w$ ,  $x$ ,  $y$ ,  $z$ .

???	???	???	???	F
	0		0	0
1				0
0	1			0

### Решение

---

```
print("x y z w F")

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

---

Вывод:

```
x y z w F
0 0 0 1 0
0 1 0 1 0
1 1 0 1 0
```

$w$  - третье место, так как все единицы.

$z$  - четвертое место, так как все нули.

$x$  - первое место, а  $y$  - второе, так как должна быть строка 0110 (третья).

**Ответ:**  $x y w z$

### №2.2 (от 8.06)

Миша заполнял таблицу истинности логической функции  $F$

$$(y \rightarrow \neg(x \rightarrow z)) \vee w$$

Но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w$ ,  $x$ ,  $y$ ,  $z$ .



???	???	???	???	F
	0			0
0	1			0
1			0	0

## Решение

---

```
print("x y z w F")

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

---

Вывод:

```
x y z w F
0 1 0 0 0
0 1 1 0 0
1 1 1 0 0
```

$y$  - третье место, так как все единицы.

$w$  - четвертое место, так как все нули.

$x$  - первое место, а  $z$  - второе, так как должна быть строка 0110 (третья).

**Ответ:**  $xzyw$

## Задача 3

### Подсказка

Выручка – сумма «Продажи»

Доход – сумма «Продажи»

Прибыль – сумма «Продажи» МИНУС сумма «Поступления»

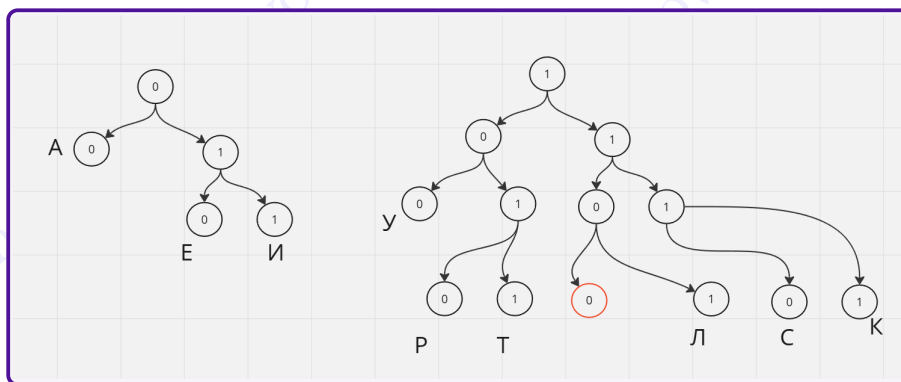
## Задача 4

### №4.1

По каналу связи передаются зашифрованные сообщения, содержащие только десять букв: А, Б, Е, И, К, Л, Р, С, Т, У; для передачи используется неравномерный двоичный код. Для девяти букв используются кодовые слова:

Буква	Кодовое слово	Буква	Кодовое слово
А	00	Л	1101
Б		Р	1010
Е	010	С	1110
И	011	Т	1011
К	1111	У	100

Укажите кратчайшее кодовое слово для буквы Б, при котором код будет удовлетворять условию Фано. Если таких кодов несколько, укажите код с наименьшим числовым значением.



**Ответ:** 1100

## Задача 5

### №5.1

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 сумма цифр в двоичной записи числа четная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 10;
  - б) если сумма цифр в двоичной записи числа нечетная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа  $R$ . Укажите максимальное число  $N$ , после обработки которого с помощью этого алгоритма получается число  $R$ , меньшее, чем 35. В ответе запишите это число в десятичной системе счисления.

### Решение

```
mx = 0
for n in range(1, 1000):
    bn = bin(n)[2:]
    if bn.count("1") % 2 == 0:
        bn = bn + "0"
        bn = "10" + bn[2:]
    else:
        bn = bn + "1"
        bn = "11" + bn[2:]
    r = int(bn, 2)
    if r < 35:
        mx = max(mx, n)
print(mx)
```

**Ответ:** 24

## №5.2 (от 8.06)

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

1. Строится двоичная запись числа  $N$ .
2. К этой записи дописываются ещё несколько разрядов по следующему правилу:
  - а) если  $N$  чётное, то к двоичной записи слева дописывается 10;
  - б) если  $N$  нечетное, то к нему справа приписываются два нуля, а слева единица;

Полученная таким образом запись (в ней как минимум на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ .

Укажите максимальное  $R$ , при условии, что  $N$  не больше 12.

### Решение

---

```
mx = 0
for n in range(1,13):
    bn = bin(n)[2:]
    if n % 2 == 0:
        bn = "10" + bn
    else:
        bn = "1" + bn + "00"
    r = int(bn, 2)
    mx = max(mx, r)
print(mx)
```

---

**Ответ:** 108

## Задача 6

### №6.1

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 6 команд: **Поднять хвост**, означающая переход к перемещению без рисования; **Опустить хвост**, означающая переход в режим рисования; **Вперёд  $n$**  (где  $n$  - целое число) вызывающая передвижение Черепахи на  $n$  единиц в том направлении, куда указывает её голова; **Назад  $n$**  (где  $n$  - целое число), вызывающая передвижение в противоположном голове направлению; **Направо  $m$**  (где  $m$  - целое число), вызывающая изменение направления движения на  $m$  градусов по часовой стрелке, **Налево  $m$**  (где  $m$  - целое число). вызывающая изменение направления движения на  $m$  градусов против часовой стрелки.

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

Черепахе был дан для исполнения следующий алгоритм:

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

**Поднять хвост**

**Вперёд 8 Направо 90 Вперёд 7 Налево 90**

**Опустить хвост**

**Повтори 4 [Вперёд 67 Направо 90 Вперёд 98 Направо 90]**

Определите площадь пересечения фигур, нарисованных при помощи алгоритма.

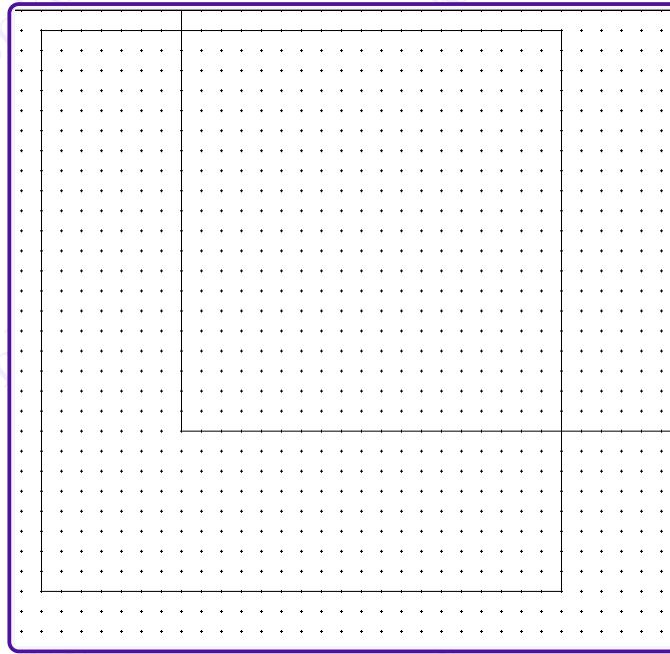
## Решение

---

```
from turtle import *

tracer(0)
screensize(400, 400)
m = 20
left(90)
pendown()
for i in range(4):
    forward(28 * m)
    right(90)
    forward(26 * m)
    right(90)
penup()
forward(8 * m)
right(90)
forward(7 * m)
left(90)
pendown()
for i in range(4):
    forward(67 * m)
    right(90)
    forward(98 * m)
    right(90)
penup()
for x in range(-100, 100):
    for y in range(-100, 100):
        goto(x * m, y * m)
        dot(3)
done()
```

---



$$20 \cdot 19 = 380$$

**Ответ:** 380

### №6.2 (от 8.06)

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 6 команд: **Поднять хвост**, означающая переход к перемещению без рисования; **Опустить хвост**, означающая переход в режим рисования; **Вперёд  $n$**  (где  $n$  – целое число), вызывающая передвижение Черепахи на  $n$  единиц в том направлении, куда указывает её голова; **Назад  $n$**  (где  $n$  – целое число), вызывающая передвижение в противоположном голове направлению; **Направо  $m$**  (где  $m$  – целое число), вызывающая изменение направления движения на  $m$  градусов по часовой стрелке, **Налево  $m$**  (где  $m$  – целое число), вызывающая изменение направления движения на  $m$  градусов против часовой стрелки.

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 10 Направо 90 Вперёд 18 Направо 90]**

**Поднять хвост**

**Вперёд 5 Направо 90 Вперёд 7 Налево 90**

**Опустить хвост**

**Повтори 2 [Вперёд 10 Направо 90 Вперёд 7 Направо 90]**

Определите, сколько точек с целочисленными координатами будут находиться внутри объединения фигур, ограниченных заданными алгоритмом линиями, включая точки на линиях.

### Решение

---

```
from turtle import *

tracer(0)

m = 15
for x in range(2):
    forward(10 * m)
    right(90)
    forward(18 * m)
    right(90)

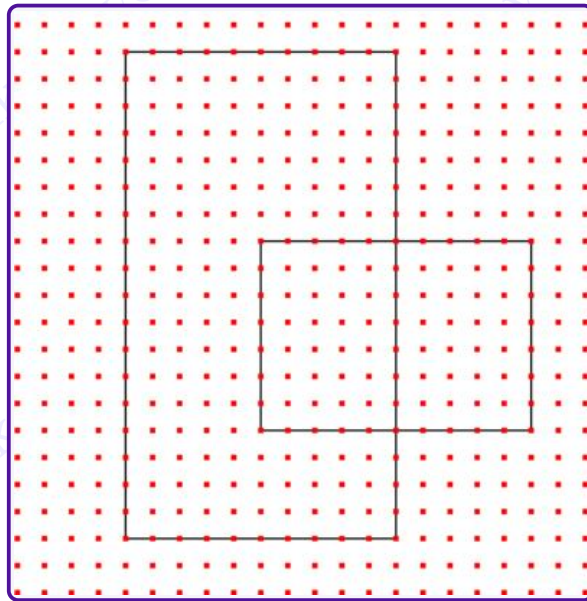
up()
forward(5 * m)
right(90)
forward(7 * m)
left(90)
down()
for x in range(2):
    forward(10 * m)
    right(90)
    forward(7 * m)
    right(90)
```



```
up()
for x in range(-50, 50):
    for y in range(-50, 50):
        goto(x * m, y * m)
        dot(4, 'red')
```

```
update()
done()
```

---



$$11 \cdot 19 + 5 \cdot 8 = 249$$

**Ответ:** 249

## Задача 7

### №7.1

Прибор автоматической фиксации нарушений правил дорожного движения делает цветные фотографии размером  $1024 \times 768$  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.

### Решение

$N = 4096$ ,  $2^{12} = 4096$ , тогда  $i = 12$ . Отсюда получаем

$$S = \frac{1024 \cdot 768 \cdot 12 \cdot 256}{2^{23}} = 288$$

**Ответ:** 288

### №7.2 (от 8.06)

Прибор автоматической фиксации нарушений правил дорожного движения делает цветные фотографии размером  $1024 \times 960$  пикселей, используя палитру из 8192 цветов. Снимки сохраняются в памяти камеры, группируются в пакеты по несколько штук, а затем передаются в центр обработки информации со скоростью передачи данных 1 474 560 бит/с. Каково максимально возможное число снимков в одном пакете, если на передачу одного пакета отводится не более 280 секунд?

### Решение

Вес одного пакета:  $1474560 \cdot 280 = 412876800$  бит

$N = 8192$ ,  $2^{13} = 8192$ , тогда  $i = 13$ . Отсюда получаем вес одной фотографии:  
 $1024 \cdot 960 \cdot 13 = 12779520$  бит

Тогда количество снимков (округляем вниз):

$$\frac{412876800}{12779520} = 32,31$$

**Ответ:** 32

## Задача 8

### №8.1

Определите количество 9-ричных 6-значных чисел, которые не начинаются с нечетных цифр, не оканчиваются цифрами 2 или 3, содержат не менее двух цифр 1.

### Решение

---

```
alph = "012345678"

count = 0
for i in "2468":
    for j in alph:
        for k in alph:
            for p in alph:
                for q in alph:
                    for r in "0145678":
                        s = i+j+k+p+q+r
                        if s.count("1") >= 2:
                            count += 1

print(count)
```

---

**Ответ:** 19868

### №8.2 (от 8.06)

Определите количество 12-ричных шестизначных чисел, в записи которых ровно одна цифра 7 и не более трёх цифр с числовым значением, превышающих 9.

## Решение

---

```
# решение 1
from itertools import product
cnt = 0
for i in product("0123456789AB", repeat = 6):
    if i.count("7") == 1 and i.count("A") + i.count("B") <= 3 and i[0] != '0':
        cnt += 1
print(cnt)

# решение 2
count = 0
for p1 in '123456789AB': # Без нуля
    for p2 in '0123456789AB':
        for p3 in '0123456789AB':
            for p4 in '0123456789AB':
                for p5 in '0123456789AB':
                    for p6 in '0123456789AB':
                        s = p1 + p2 + p3 + p4 + p5 + p6
                        if s.count('7') == 1:
                            if s.count('A') + s.count('B') <= 3:
                                count += 1
print(count)
```

---

**Ответ:** 888669

### №8.3 (от 8.06)

Все пятибуквенные слова, составленные из букв К, О, М, П, Ъ, Т, Е, Р, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. EEEEE
2. EEEЕК
3. EEEЕМ
4. EEEEEО
5. EEEEEП
6. EEEЕР
7. EEEЕТ
8. EEEЕЪ

Под каким номером в списке стоит последнее слово, которое не содержит букв К и содержит ровно две буквы Р?

#### Решение

---

```
a = 'ЕКМОПРТЪ'
c = 0
res = 0
for x1 in a:
    for x2 in a:
        for x3 in a:
            for x4 in a:
                for x5 in a:
                    c += 1
                    s = x1+x2+x3+x4+x5
                    if s.count('K') == 0 and s.count('P') == 2:
                        res = c
print(c)
```

---

**Ответ:** 32768

## Задача 9

### №9.1

Файл электронной таблицы, в каждой строке 4 натуральных числа. Определите количество строк в таблице, содержащих числа, для которых выполнены оба условия:

- наибольшее из 4 чисел меньше суммы трёх других
- все четыре числа различны

### Решение

Для проверки первого условия можно использовать формулу:

=ЕСЛИ(МАКС(A1:D1) < СУММ(A1:D1)-МАКС(A1:D1); 1; 0)

Для проверки второго условия используем

=СЧЕТЕСЛИ(A1:D1;A1)

=СЧЕТЕСЛИ(A1:D1;B1)

=СЧЕТЕСЛИ(A1:D1;C1)

=СЧЕТЕСЛИ(A1:D1;D1)

Затем проверяем что все значения 1:

=ЕСЛИ(И(E1=1;F1=1;G=1;H1=1);1;0)

### №9.2 (от 8.06)

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

- в строке есть только одно число, которое повторяется трижды, остальные три числа без повторений;
- повторяющееся число строки не меньше, чем среднее арифметическое трёх её неповторяющихся чисел.

В ответе запишите только число.

## Решение

Для проверки первого условия используем

=СЧЕТЕСЛИ(A1:F1;A1)

=СЧЕТЕСЛИ(A1:F1;B1)

=СЧЕТЕСЛИ(A1:F1;C1)

=СЧЕТЕСЛИ(A1:F1;D1)

=СЧЕТЕСЛИ(A1:F1;E1)

=СЧЕТЕСЛИ(A1:F1;F1)

Затем проверяем что есть 3 значения равных 3, а остальные равны 1:

=ЕСЛИ(И(СЧЕТЕСЛИ(G1:L1;3)=3;СЧЕТЕСЛИ(G1:L1;1)=3);1;0)

Для проверки второго условия найдем сначала среднее значение неповторяющихся и отдельно повторяющиеся:

=СУММЕСЛИ(G1:L1;1;A1:F1)/3 (ячейка M1)

=СУММЕСЛИ(G1:L1;3;A1:F1)/3 (ячейка N1)

А затем сравним:

=ЕСЛИ(N1>=M1;1;0)

## №9.3 (от 8.06)

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите количество строк таблицы, для чисел которых выполнены оба условия:

- в строке есть ровно одно число, которое повторяется трижды, и остальные числа без повторений;
- квадрат суммы всех повторяющихся чисел строки больше квадрата суммы всех неповторяющихся чисел строки

## Решение

Для проверки первого условия используем

=СЧЕТЕСЛИ(A1:F1;A1)

=СЧЕТЕСЛИ(A1:F1;B1)

=СЧЕТЕСЛИ(A1:F1;C1)

=СЧЕТЕСЛИ(A1:F1;D1)

=СЧЕТЕСЛИ(A1:F1;E1)

=СЧЕТЕСЛИ(A1:F1;F1)

Затем проверяем что есть 3 значения равных 3, а остальные равны 1:

=ЕСЛИ(И(СЧЕТЕСЛИ(G1:L1;3)=3;СЧЕТЕСЛИ(G1:L1;1)=3);1;0)

Для проверки второго условия найдем сначала сумму неповторяющихся:

=СУММЕСЛИ(G1:L1;1;A1:F1) (ячейка M1)

И отдельно сумму повторяющегося:

=СУММЕСЛИ(G1:L1;3;A1:F1) (ячейка N1)

А затем сравним:

=ЕСЛИ(N1\*N1>M1\*M1;1;0)

## Задача 10

### №10.1

С помощью текстового редактора определите, сколько раз встречается сочетание букв «рук» или «Рук» в тексте глав IV, V, VI и VII второй части тома 2 романа Л.Н. Толстого «Война и мир». В ответе укажите только число.

### Решение

Удобнее всего будет удалить из файла все главы кроме IV, V, VI и VII, после чего с помощью стандартного Ctrl + F найти сначала «рук», а потом «Рук» в оставшемся тексте.

## Задача 11

### №11.1

В некоторой стране автомобильный номер длиной 6 символов составляют из заглавных букв (используются только 33 различных буквы) и десятичных цифр в любом порядке. Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байтов (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов). Определите объём памяти, отводимый этой программой для записи 125 номеров. (Ответ дайте в байтах.)



### Решение

33 буквы и 10 цифр, значит  $i = 6$  бит.

$$\frac{6 \cdot 6}{8} = 5 \text{ байт.}$$

$$125 \cdot 5 = 625 \text{ байт.}$$

**Ответ:** 625

### №11.2 (от 8.06)

На предприятии каждой изготовленной детали присваивают серийный номер длиной 261 символ. Для его хранения отведено одинаковое и минимально возможное число байт. При этом используется посимвольное кодирование серийных номеров, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 862 серийных номеров отведено не более 276 Кбайт памяти. Определите максимально возможную мощность алфавита, из которого составляются серийные номера

### Решение

276 Кбайт на 862 номера, значит на один номер  $\frac{276 \cdot 2^{13}}{862} = 2622.9$ .

Округляем в меньшую сторону, иначе не хватит выделенного места.

2622 бита на 1 номер, значит на один символ  $\frac{2622}{261} = 10.04$

Округляем в меньшую сторону, иначе не хватит выделенного места.

Значит ответ:  $2^{10} = 1024$

**Ответ:** 1024

### №11.3 (от 8.06)

Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного наблюдения является целое число от 0 до 100%, записываемое при помощи минимально возможного количества бит. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.

### Решение

$100 < 2^7$ , значит 7 бит на одну запись.

$$\frac{7 \cdot 80}{8} = 70 \text{ Байт}$$

**Ответ:** 70

## №11.4 (от 8.06)

При регистрации в компьютерной системе каждому объекту присваивается идентификатор, состоящий из 317 символов и содержащий только десятичные цифры и символы из 4090-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 Мбайт), необходимый для хранения 262144 идентификаторов. В ответе запишите только целое число - количество Мбайт.

### Решение

Для кодирования 4100 символов необходимо 13 бит. Тогда для кодирования идентификатора необходимо

$$\frac{317 \cdot 13}{8} = 516 \text{ Байт}$$

Для хранения 262144 идентификаторов необходимо

$$\frac{516 \cdot 262144}{2^{20}} = 129 \text{ Мбайт}$$

**Ответ:** 129

## Задача 12

### №12.1

Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 108 идущих подряд цифр 7? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (33333) ИЛИ нашлось (777)

ЕСЛИ нашлось (33333)

ТО заменить (33333,7)

ИНАЧЕ заменить (777, 3)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

## Решение

---

```
s = '7'*108
while '33333' in s or '777' in s:
    if '33333' in s:
        s = s.replace('33333', '7', 1)
    else:
        s = s.replace('777', '3', 1)
print(s)
```

---

**Ответ:** 3337

### №12.2 (от 8.06)

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 84 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (1111) ИЛИ нашлось (8888)

ЕСЛИ нашлось (1111)

ТО заменить (1111, 8)

ИНАЧЕ заменить (8888, 11)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

## Решение

---

```
s = '8'*84
while '1111' in s or '8888' in s:
    if '1111' in s:
        s = s.replace('1111', '8', 1)
    else:
        s = s.replace('8888', '11', 1)
print(s)
```

---

Ответ: 11888

## Задача 13

### №13.1

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 106.184.0.0 и маской сети 255.248.0.0. Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса не кратна 2?

## Решение

---

```
from ipaddress import ip_network
cnt = 0
for ip in ip_network('106.184.0.0/255.248.0.0'):
    new_ip = bin(int(ip))[2:].zfill(32)
    if new_ip.count('1') % 2 != 0:
        cnt += 1
print(cnt)
```

---

Решение руками:

Переведем в 2-ю СС IP адрес сети и маску.

NET: 01101010.10111000.00000000.00000000

MASK: 11111111.11111000.00000000.00000000

Значит все IP адреса в этой сети будут иметь вид:

01101010.10111xxx.xxxxxxxxxx.xxxxxxxxxx

В исходной строке 8 единиц, для того чтобы их количество было нечетным на свободные позиции нужно поставить нечетное количество единиц, то есть 19, 17, 15, ..., 5, 3, 1.

Для каждого возможного количества единиц необходимо посчитать количество сочетаний по формуле:

$$C_n^k = \frac{n!}{(n-k)!k!}$$

где  $n = 19$ , а  $k = 19, 17, 15, \dots, 5, 3, 1$ .

Или сгенерируем последнюю часть при помощи программы и посчитаем кол-во единиц.

---

```
from itertools import product
```

```
cnt = 0
```

```
for ip in product('01', repeat=19):
```

```
    new_ip = '01101010.10111' + ''.join(ip)
```

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

```
        cnt += 1
```

```
print(cnt)
```

---

**Ответ:** 262144

### №13.2 (от 8.06)

Два узла, находящиеся в одной сети, имеют IP-адреса 115.127.30.120 и 115.127.151.120. Укажите наибольшее возможное значение третьего слева байта маски сети. Ответ запишите в виде десятичного числа.

### Решение

Переведем в 2-ю СС IP адреса.

IP 1: 01110011.01111111.00011110.01111000

IP 2: 01110011.01111111.10010111.01111000

Видим, что самый первый бит третьего байта маски отличаются, значит, чтобы адреса были в одной сети, нужно чтобы здесь был 0, значит и весь третий байт это 0.

**Ответ:** 0

### №13.3 (от 8.06)

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая - к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места - нули. Обычно маска записывается по тем же правилам, что и IP-адрес, - в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Например, если IP-адрес узла равен 131.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 131.32.240.0.

Для узла с IP адресом 170.155.137.181 адрес сети равен 170.155.136.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

### Решение

Переведем в 2-ю СС IP адрес и сеть.

IP1: 10101010.10011011.10001001.10110101

NET: 10101010.10011011.10001000.00000000

Видим, что в третьем байте у IP и NET отличаются лишь последние биты, поэтому можем взять третий байт как 11111110, что в десятичной

**Ответ:** 254

### №13.4 (от 8.06)

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 112.160.0.0 и сетевой маской 255.240.0.0.

Сколько в этой сети IP-адресов, для которых количество единиц в двоичной записи IP-адреса не кратно 5?

В ответе укажите только число.

#### Решение

Решение руками + перебор комбинаций в проге:

Переведем в 2-ю СС IP адрес сети и маску.

NET: 01110000.10100000.00000000.00000000

MASK: 11111111.11110000.00000000.00000000

Значит все IP адреса в этой сети будут иметь вид:

01110000.1010xxxx.xxxxxxxxxx.xxxxxxxxxx

В исходной строке 5 единиц.

Сгенерируем в программе все возможные комбинации для неизвестной части ip адреса. Найдем для каждой комбинации кол-во единиц в полученном ip и проверим, кратно ли оно 5:

---

```
import itertools

c = 0
l = itertools.product([0, 1], repeat=20)

for i in l:
    if (sum(i)+5) % 5 != 0:
        c += 1

print(c)
```

---

**Ответ:** 832810

## Задача 14

### №14.1

Значение арифметического выражения

$$3 \cdot 289^{2024} + 81 \cdot 49^{121} - 9 \cdot 16^{81} - 6011$$

записали в системе счисления с основанием 31. Определите сумму цифр с числовым значением, не превышающим 17, в записи этого числа.

### Решение

*# Решение 1*

```
s = 3 * 289**2024 + 81 * 49**121 - 9 * 16**81 - 6011
```

```
sm = 0
```

```
while s > 0:
```

```
    digit = s%31 # Числовое значение текущей цифры
```

```
    if digit <= 17:
```

```
        sm += digit
```

```
    s //= 31
```

```
print(sm)
```

*# Решение 2*

```
s = 3 * 289**2024 + 81 * 49**121 - 9 * 16**81 - 6011
```

```
# Список для хранения количества цифр в числе в записи в 31-ричной СС
```

```
digits = [0] * 31
```

```
while s > 0:
```

```
    digits[s % 31] += 1
```

```
    s //= 31
```

```
sm = 0
```

```
# Перебор цифр с числовым значением, не превышающим 17
```

```
for i in range(18):
```

```
    sm += digits[i]*i
```

```
print(sm)
```

**Ответ:** 16750



## №14.2 (от 8.06)

Значение арифметического выражения  $3^{100} - x$ , где  $x$  - целое положительное число, не превышающее 2030, записали в троичной системе счисления. Определите наименьшее значение  $x$ , при котором в троичной записи числа, являющегося значением данного арифметического выражения, содержится ровно два нуля.

В ответе запишите число в десятичной системе счисления.

### Решение

```
def tri(n):
    s = ''
    while n > 0:
        s = str(n % 3) + s
        n //= 3
    return s

for x in range(1, 2030 + 1):
    if tri(3 ** 100 - x).count('0') == 2:
        print(x)
        break
```

**Ответ:** 9

## Задача 15

### №15.1

Для какого наименьшего натурального числа  $A$  логическое выражение истинно при любом целом положительном значении переменной  $x$ .

$$(\text{ДЕЛ}(x, 2) \rightarrow \overline{\text{ДЕЛ}(x, 5)}) \vee (x + A \geq 70)$$

## Решение

```
for A in range(1, 1000):
    flag = 1
    for x in range(1, 10000):
        F = ((x % 2 == 0) <= (x % 5 != 0)) or (x + A >= 70)
        if not F:
            flag = 0
            break
    if flag == 1:
        print(A)
        break
```

Ответ: 60

## №15.2 (от 8.06)

Для какого наибольшего натурального числа Логическое выражение истинно при любом целом положительном значении переменной  $x$ , если  $B = [70, 90]$ .

$$\text{ДЕЛ}(x, A) \vee ((x \in B) \rightarrow \overline{\text{ДЕЛ}(x, 27)})$$

## Решение

```
B = list(range(70, 90 + 1))
for A in range(1, 1000):
    flag = 1
    for x in range(1, 10000):
        F = (x % A == 0) or ((x in B) <= (not(x % 27 == 0)))
        if not F:
            flag = 0
            break
    if flag == 1:
        print(A)
```

Ответ: 81

### №15.3 (от 8.06)

Для какого наибольшего натурального числа Логическое выражение истинно при любом целом положительном значении переменной  $x$ , если  $B = [70, 90]$ .

$$\text{ДЕЛ}(x, A) \vee ((x \in B) \rightarrow \overline{\text{ДЕЛ}(x, 22)})$$

#### Решение

---

```
B = list(range(70, 90 + 1))

for A in range(1, 1000):
    flag = 1
    for x in range(1, 10000):
        F = (x % A == 0) or ((x in B) <= (not(x % 22 == 0)))
        if not F:
            flag = 0
            break
    if flag == 1:
        print(A)
```

---

**Ответ:** 88

## Задача 16

### №16.1

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

$$F(n) = 1 \text{ при } n = 1$$

$$F(n) = 2 * n * F(n - 1), \text{ если } n > 1$$

Чему равно значение выражения  $(F(2024) - 4 * F(2023))/F(2022)$ ?

## Решение

```
# решение 1
import sys
sys.setrecursionlimit(10**6)
def F(n):
    if n == 1:
        return 1
    elif n > 1:
        return 2 * n * F(n - 1)
print((F(2024) - 4 * F(2023)) // F(2022))
```

```
# решение 2
F = [0] * 2500
for n in range(1, 2025):
    if n == 1:
        F[n] = 1
    elif n > 1:
        F[n] = 2 * n * F[n - 1]
print((F[2024] - 4 * F[2023]) // F[2022])
```

**Ответ:** 16362024

### №16.2 (от 8.06)

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

$$F(n) = 1 \text{ при } n = 1$$

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

Чему равно значение выражения  $(F(2024) / 4 - F(2023)) / F(2022)$ ?

## Решение

```
# решение 1
import sys
sys.setrecursionlimit(10**6)
def F(n):
    if n == 1:
        return 1
    elif n > 1:
        return n * F(n - 1)
print((F(2024) // 4 - F(2023))//F(2022))
```

```
# решение 2
F = [0] * 2500
for n in range(1, 2025):
    if n == 1:
        F[n] = 1
    elif n > 1:
        F[n] = n * F[n - 1]
print((F[2024] // 4 - F[2023])//F[2022])
```

**Ответ:** 1021615

## Задача 17

### №17.1

В файле содержится последовательность натуральных чисел. Элементы последовательности могут принимать целые значения от 1 до 100 000 включительно.

Определите количество пар элементов последовательности, в которых сумма остатков от деления обоих элементов на 18 равна минимальному элементу последовательности. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

## Решение

---

```
f = open("17.txt")
elements = [int(s) for s in f]
min_element = min(elements)
pairs = []
for i in range(1, len(elements)):
    if elements[i] % 18 + elements[i - 1] % 18 == min_element:
        pairs.append(elements[i] + elements[i - 1])
print(len(pairs), max(pairs))
```

---

### №17.2 (от 8.06)

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -100 000 до 100 000 включительно. Определите количество пар элементов последовательности, в которых хотя бы одно отрицательное число, а сумма элементов пары меньше, чем количество чисел из последовательности, кратных 32. В ответе запишите без пробелов количество найденных пар чисел, затем максимальную из сумм элементов таких пар. Под парой элементов подразумеваются два соседних элемента.

## Решение

---

```
f = open("17.txt")
a = [int(x) for x in f]
cnt_32 = 0
for i in range(len(a)):
    if a[i] % 32 == 0:
        cnt_32 += 1
ans = []
for i in range(len(a) - 1):
    if (a[i] < 0 or a[i + 1] < 0) and ((a[i] + a[i + 1]) < cnt_32):
        ans.append([a[i] + a[i + 1]])
print(len(ans), max(ans))
```

## Задача 18

### №18.1

Квадрат разлинован 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 «угловых» 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  $N \times N$ , каждая ячейка которой соответствует клетке квадрата. Внутренние и внешние стены обозначены утолщёнными линиями.

### Решение

**Решение приведено для идентичного прототипа из базы задач «Школково» в виду отсутствия файла**

Нам дано поле 18 на 18, создадим еще одно поле ниже (A20:R37), в котором будем считать искомые значения. В ячейку A20 запишем число из A1 без изменений; Заполняем всю остальную таблицу аналогично самым простым задачам. В ячейку B20 вставляем формулу:

$$=A20+B1$$

Растягиваем её для первой строчки до R20. В ячейку A21 вставляем формулу:

$$=A20+A2$$

Растягиваем её для первого столбца до A37. В ячейку B21 вставляем формулу:

$$=МАКС(A21;B20)+B2$$

Растягиваем её до конца таблицы - до ячейки R37.

Выделяем желтым цветом диапазон ячеек, которые стоят справа от стены и зеленым цветом, которые стоят под стеной. В желтые ячейки можно прийти только из ячеек, находящихся сверху, поэтому модернизируем формулу. В желтые ячейки необходимо написать формулу, аналогичную формуле для первого столбца, то есть исходная ячейка + предыдущая сверху. В зеленые ячейки можно прийти только из ячеек, находящихся слева, поэтому модернизируем формулу. В зеленые ячейки необходимо написать формулу, аналогичную формуле для первой строки, то есть исходная ячейка + предыдущая слева.

Важно отметить, что так как в этой задаче конечных ячеек несколько, максимальную и минимальную суммы нужно выбирать из всех возможных вариантов.

78	128	148	221	183	110	19	-76	-174	-98	-173	-227	-271	-298	-231	-240	-224	-273
82	41	228	239	249	264	339	270	294	269	228	325	373	421	381	462	478	515
-7	31	182	285	273	173	418	446	515	507	410	479	483	404	367	508	586	645
-40	71	255	295	237	173	415	512	585	646	745	735	665	751	421	600	577	687
-96	165	231	216	180	217	477	552	659	733	658	669	698	841	514	501	661	724
-55	217	196	292	221	247	392	535	582	754	668	613	738	826	548	461	695	686
-114	162	178	314	310	321	312	467	623	714	700	719	666	914	815	894	950	738
-144	63	89	340	251	303	370	447	685	721	696	779	742	912	969	885	948	703
-78	153	141	388	305	296	466	532	624	646	662	845	826	938	891	906	898	684
-169	155	121	485	418	509	589	659	741	647	628	930	832	851	903	963	1050	979
-225	219	125	557	482	411	574	656	785	713	658	879	857	879	826	893	1109	1181
-304	130	73	605	539	606	601	590	693	772	738	913	1000	1075	1021	1002	1182	1176
-219	66	3	647	552	654	587	680	684	823	825	839	1042	1057	1019	1062	1134	1090
-176	8	-12	704	698	690	698	636	759	921	885	908	972	1037	1085	1119	1119	1219
-227	-57	-18	691	763	836	860	839	787	948	882	946	935	1093	1143	1134	1055	1169
-172	-47	-94	647	731	737	908	811	735	653	725	698	1006	1009	1090	1106	1060	1094
-223	-22	9	648	660	734	898	826	787	751	833	835	1023	985	1110	1107	1157	1245
-145	25	117	704	728	725	887	829	729	715	931	865	1114	1045	1107	1172	1086	1215

Для того, чтобы найти минимальную сумму необходимо заменить во всех формулах МАКС на МИН.

**Ответ:** 1215 -1299



## Задача 19-21

### №19-21.1

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.

#### № 19

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 хода Вани.

Найденные значения запишите в ответе в порядке возрастания.

#### № 21

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

---

```
from functools import lru_cache

@lru_cache()
def F(s):
    if s >= 129:
        return 0
    t = [F(s+1), F(s*2)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n) + 1
    return -max(t)
```

#19

```
for i in range(1, 129):
    if F(i) == -1:
        print(i)
print()
```

#20

```
for i in range(1, 129):
    if F(i) == 2:
        print(i)
print()
```

#21

```
for i in range(1, 129):
    if F(i) == -2:
        print(i)
```

---

**Ответ:** 19) 64 20) 32 63 21) 62

## №19-21.2 (от 8.06)

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 или четыре камня либо увеличить количество камней в куче в три раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 19 или 75 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 43.

Победителем считается игрок, сделавший последний ход, т. е. первым получивший кучу, в которой будет 43 или больше камней. В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 42$ .

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.

### № 19

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

Найдите два таких 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.

### № 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, позволяющей гарантированно выиграть первым ходом.

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

## Решение

---

```
from functools import lru_cache

@lru_cache()
def F(s):
    if s >= 43: return 0
    t = [F(s+1), F(s+4), F(s * 3)]
    n = [i for i in t if i <= 0]
    if n: return -max(n) + 1
    return -max(t)

#19
for i in range(1, 42 + 1):
    if F(i) == -1:
        print(i)
print()

#20
for i in range(1, 42 + 1):
    if F(i) == 2:
        print(i)
print()

#21
for i in range(1, 42 + 1):
    if F(i) == -2:
        print(i)
        break
```

---

**Ответ:** 19) 14 20) 10 13 21) 9

## Задача 22

### №22.1

**Задача не с ДВ, а просто такой же прототип из базы задач школково.**

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 первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

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

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

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $5 + 7 = 12$  мс.

## Решение

Для начала распределим значения из столбца С. Для этого выделим данные в этом столбце и с помощью кнопок «Данные»->«Текст по столбцам» сформируем их на разные столбцы.

В клетку A102(то есть в конце столбца А) в конце добавляем 0, чтобы функция ВПР не возвращала ошибку и корректно считала время независимых процессов.

В ячейку I2 поместим формулу =МАКС(F2:H2)+B2 и растянем её вниз. В ячейку F2 поместим формулу =ВПР(C2;\$A\$1:\$I\$102;9;0) и растянем её на диапазон F2:H101. В ячейку J2 запишем формулу: =МАКС(I2:I101). Это значение и будет являться ответом.

## Задача 23

### №23.1

Исполнитель преобразует число на экране. У исполнителя есть три команды, которым присвоены номера:

- А. Вычти 1
- В. Вычти 2
- С. Найди целую часть от деления на 3

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 16 результатом является число 6, при этом траектория вычислений содержит число 11?

## Решение

---

```
def f(a, b, r=False):
    if a == b and r: return 1
    if a < b: return 0
    if a == 11: r = True
    return f(a - 1, b, r) + f(a - 2, b, r) + f(a // 3, b, r)
print(f(16, 6))
```

---

**Ответ:** 64

## №23.2 (от 8.06)

Исполнитель преобразует число на экране. У исполнителя есть две команды, которым присвоены номера:

Прибавить 1

Прибавить 2

Прибавить 3

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 35, при этом траектория вычислений содержит число 7?

### Решение

---

```
from functools import lru_cache

@lru_cache(None)
def f(a, b, r=False):
    if a == b and r:
        return 1
    if a > b:
        return 0
    if a == 7:
        r = True
    return f(a + 1, b, r) + f(a + 2, b, r) + f(a + 3, b, r)
print(f(1, 35))
```

---

**Ответ:** 381662184

## Задача 24

### №24.1

Текстовый файл состоит из заглавных букв латинского алфавита А, В, С, D, Е и F. Определите максимальное количество идущих подряд символов в прилагаемом файле, среди которых пара символов DE (в указанном порядке) встречается не более 240 раз. Для выполнения этого задания следует написать программу.

### Решение

---

```
f = open(...)
s = f.read()
s = s.replace('DE', 'D E').split()
mx = -1
for i in range(len(s)):
    r = ''.join(s[i:i + 241])
    mx = max(mx, len(r))
print(mx)
```

---

### №24.2 (от 8.06)

Найти подпоследовательность максимальной длины, которая имеет арифметический смысл и содержит только цифры 6, 7, 8, 0 и знаки операций «-», «\*». При этом должны выполняться следующие условия:

Возле цифры 0 не может быть знака операции.

В выражении не должно быть умножения на отрицательные числа.

В выражении не должно быть повторяющихся знаков операций, таких как "\*\*\*".

Пример правильного выражения: 6787-86

Пример неправильного выражения: 6786\*-78 (содержит повторяющиеся знаки операции и отрицательное число)



## Решение

---

```
s = open("24.txt").readline()

comb = [('--', '_-'), ('**', '*_*'),
        ('-0', '_-'), ('*0', '*_'),
        ('0-', '_-'), ('0*', '*_'),
        ('-*', '_-*'), ('*-', '*_')]

for pair in comb:
    while pair[0] in s: # Пока есть неподходящее сочетание
        s = s.replace(pair[0], pair[1]) # Заменяем его нужное сочетание

s = s.split('_-')

print(s)

mx = -1

for i in s:
    if len(i) > 0 and i[-1] in '*-':
        i = i[:-1]
    if len(i) > 0 and i[0] == '*':
        i = i[1:]
    while len(i) > 0 and i[0] == '0':
        i = i[1:]

    if "*" in i or "-" in i:
        mx = max(mx, len(i))

print(mx)
```

---

## Задача 25

### №25.1

Напишите программу, которая перебирает целые числа, большие 600000 в порядке возрастания и ищет среди них такие, у которых есть натуральный делитель, оканчивающийся на цифру 7, и не равный самому числу и числу 7. В ответе запишите в первой строке таблицы первые пять найденных чисел в порядке возрастания, а во втором столбце - наименьший делитель для каждого из них, оканчивающихся цифрой 7, неравный ни самому числу, ни числу 7. Кол-во строк в таблице избыточно.

### Решение

---

```
def div(n):
    divs = set()
    for i in range(1, int(n**0.5)+1):
        if n%i==0:
            divs.add(i)
            divs.add(n//i)
    return sorted(divs)

k = 0
for i in range(600_001, 1_000_000):
    divs = div(i) # Делители числа i
    for d in divs:
        if d%10==7 and d != 7 and d != i:
            print(i, d)
            k += 1
            break
    if k == 5:
        break
```

---

**Ответ:** 600001 437 600002 47 600003 1227 600005 217 600012 16667

## №25.2 (от 8.06)

Пусть  $M$  - сумма минимального и максимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то считаем значением равным нулю.

Напишите программу, которая перебирает целые числа, большие 700 000, в порядке возрастания и ищет среди них такие, для которых  $M$  оканчивается на 4. В ответе запишите в первом столбце таблицы первые пять найденных чисел в порядке возрастания, а во втором столбце - соответствующие им значения  $M$ . Например, для числа 20  $M = 2 + 10 = 12$ . Количество строк в таблице для ответа избыточно.

### Решение

---

```
def div(n):
    divs = set()
    for i in range(2, int(n**0.5)+1):
        if n%i==0:
            divs.add(i)
            divs.add(n//i)
    return sorted(divs)

k = 0
for i in range(700_001, 1_000_000):
    divs = div(i) # Делители числа i
    if len(divs) != 0:
        M = divs[0] + divs[-1]
        if M % 10 == 4:
            print(i, M)
            k += 1
        if k == 5:
            break
```

---

**Ответ:** 700004 350004 700009 41194 700023 233344 700024 350014 700044 350024

## Задача 26

### №26.1

При онлайн-покупке билета на концерт известно, какие места в зале уже заняты. Необходимо купить билет на такое место в ряду, чтобы перед ним как можно больше идущих подряд кресел с таким же номером было свободно. Если места, удовлетворяющие этому условию, есть в нескольких рядах, то нужно выбрать ряд, расположенный как можно ближе к сцене. В ответе запишите два целых числа: искомый номер ряда и количество свободных кресел перед выбранным местом. Нумерация рядов и мест ведётся с 1. Гарантируется, что хотя бы одно такое место в зале есть.

Входные данные

В первой строке входного файла находятся три числа:  $N$  - количество занятых мест в зале (целое положительное число, не превышающее 10 000),  $M$  - количество рядов (целое положительное число, не превышающее 100 000) и  $K$  - количество мест в каждом ряду (целое положительное число, не превышающее 100 000). В следующих  $N$  строках находятся пары натуральных чисел: номер ряда и номер места занятого кресла соответственно (первое число не превышает значения  $M$ , а второе -  $K$ ).

Выходные данные

Два целых положительных числа: искомый номер ряда и количество свободных кресел перед выбранным местом.

### Решение

---

```
File = open("26.txt")
N, M, K = map(int, File.readline().split())
places = [] # Список для считывания занятых мест
for i in range(N):
    row, place = map(int, File.readline().split())
    places.append([place, row]) # На первое место ставим место, затем ряд
    # Такое расположение обеспечивает сортировку именно по номеру места
places.sort()
# Максимальное количество свободных мест перед искомым свободным местом
```

```

maxCount = 0
# Минимальный номер ряда такого места
minRow = 0
for p in range(1, N):
    previousPlace = places[p - 1][0] # Номер прошлого места
    previousRow = places[p - 1][1] # Номер ряда прошлого места
    currentPlace = places[p][0] # Номер текущего места
    currentRow = places[p][1] # Номер ряда текущего места
    # Если номера совпадают, то текущее место стоит после прошлого
    if currentPlace == previousPlace:
        # Считаем количество свободных мест перед предыдущим местом:
        count = (currentRow - 1) - (previousRow + 1)
        # Если максимум меняется, то полностью меняем данные
        if maxCount < count:
            maxCount = count
            minRow = currentRow - 1
        # Если максимум остался, то смотрим, можно ли уменьшить номер ряда
        elif maxCount == count:
            minRow = min(minRow, currentRow - 1)

    elif currentPlace - previousPlace == 1:
        # Количество мест от конца зала до текущего места
        count = M - (previousRow + 1)
        if maxCount < count:
            maxCount = count
            minRow = M
        elif maxCount == count:
            minRow = min(minRow, M)

    # Количество мест от предыдущего места до сцены
    count = currentRow - 2
    if maxCount < count:
        maxCount = count

```

```

        minRow = currentRow - 1
    elif maxCount == count:
        minRow = min(minRow, currentRow - 1)
    else: # Есть полностью пустой ряд
        maxCount = K - 1 # Кол-во свободных мест это K-1
        minRow = M # Минимальный ряд в таком случае - последний
        # Дальше не нужно перебирать, максимальное кол-во не поменяется
        break

print(minRow, maxCount)

```

---

### Решение экселем:

Если мы знаем, что место 1 занято в рядах 1 и 6, значит между ними есть ряды 2, 3, 4, 5, которые свободны. А если они свободны, значит перед местом в 5-м ряду есть три свободных места. Найти можем как  $(6 - 1) - (1 + 1) = 3$ .

Если же это место первое занято, например в ряду 4, то перед ним всего  $4 - 1 = 3$  свободных мест, значит перед местом в 3-м ряду мест  $4 - 2 = 2$ .

Также, если это первое место, то нужно от предыдущего рассматриваемого также посчитать количество мест до конца зала, так как последнее место в «столбце» мест тоже может считаться. Тогда, если предыдущее последнее место занято в ряду 4, количество мест это  $6 - (4 + 1)$ .

Чтобы это посчитать в экселе, отсортируем по возрастанию сначала столбец *B*, потом столбец *A*.

	A	B
1	1	1
2	6	1
3	5	2
4	4	3
5	2	4
6	5	5
7	3	6
8	6	6
9	4	7

В Ячейку C2 запишем 0.

В ячейку C3 вставим формулу и растянем до конца:

$$= \text{ЕСЛИ}(B2 = B3; \text{МАКС}((A3 - 1) - (A2 + 1); C2); A3 - 2)$$

Чтобы определить, можно ли выбрать место в конце зала, в ячейку D2 запишем формулу и растянем до конца:

$$= \text{ЕСЛИ}(B3 <> B2; 6 - (A2 + 1); 0)$$

Чтобы определить максимальное кол-во мест в этом «столбце» мест с одинаковым номером, запишем в ячейку E2 формулу и растянем до конца:

$$= \text{МАКС}(C2 : D2)$$

Чтобы определить номер ряда, в ячейку F2 запишем формулу и растянем до конца:

$$= \text{ЕСЛИ}(E2 = \text{МАКС}(E : E); A2 - 1; 9999999)$$

Если текущее кол-во свободных мест — это максимум столбца - выводим номер ряда -1, иначе что-то большое.

Окончательный номер ряда узнаем, если сравним для каждого места максимальное перед ним и после него, если после – больше, тогда минимальный номер ряда это самый последний ряд. В ячейку G2 запишем и растянем формулу:

$$= \text{ЕСЛИ}(C2 \geq D2; F2; 6)$$

Таким образом нужно выбрать строчку с минимальным числом из столбца G, в ней в столбце G - первый ответ, в столбце E - второй.

	A	B	C	D	E	F	G
1	Ряды	Места	Места перед	Места после	Максимальное Кол-во мест	Минимальный Номер ряда	
2		1	1	0	0	9999999	9999999
3		6	1	3	-1	5	5
4		5	2	3	0	4	4
5		4	3	2	1	9999999	9999999
6		2	4	0	3	1	6
7		5	5	3	0	4	4
8		3	6	1	0	9999999	9999999
9		6	6	1	-1	9999999	9999999
10		4	7	2	1	9999999	9999999

## №26.2 (от 8.06)

Задание №26 При онлайн-покупке билета на концерт известно, какие места в зале уже заняты. Необходимо купить два билета на такие соседние места в одном ряду, чтобы перед ними все кресла с такими же номерами были свободны, а ряд находился как можно дальше от сцены. Если в этом ряду таких пар мест несколько, найдите пару с наибольшими номерами. В ответе запишите два целых числа: искомый номер ряда и наибольший номер места в найденной паре. Нумерация рядов и мест ведётся с 1. Гарантируется, что хотя бы одна такая пара в зале есть.

Входные данные:

В первой строке входного файла находятся три числа: N - количество занятых мест в зале (целое положительное число, не превышающее 10 000), M - количество рядов (целое положительное число, не превышающее 100 000) и K - количество мест в каждом ряду (целое положительное число, не превышающее 100 000). В следующих N строках находятся пары натуральных чисел: номер ряда и номер места занятого



кресла соответственно (первое число не превышает значения  $M$ , а второе -  $K$ ).

Выходные данные:

Два целых положительных числа: наибольший номер ряда и наибольший номер места в найденной паре кресел. Типовой пример организации данных во входном файле:

7 7 8

1 1

6 6

5 5

6 7

4 4

2 2

3 3

При таких исходных данных ответом является пара чисел 5 и 8. Условию задачи удовлетворяют места 7 и 8 в ряду 5: перед креслами 7 и 8 нет занятых мест и это последняя из двух возможных пар в этом ряду. В рядах 6 и 7 искомую пару найти нельзя.

### Идея решения

Так как нам нужны ВСЕ свободные места ПЕРЕД креслом, нужно "отфильтровать" оставив только данные по местам, которые первые в своих столбцах. То есть условно есть место 2 5 и 4 5. Перед местом 4 5 уже есть 2 5, поэтому его не берём. Потом пройтись по всем таким местам и выбрать те, которые в одном ряду и рядом.

### №26.3 (от 8.06)

Отбор абитуриентов в вуз происходит по сумме баллов трех экзаменов: по русскому языку, математике и информатике. На заранее известное количество мест зачисляются абитуриенты, набравшие большую сумму баллов по результатам трех экзаменов. Все абитуриенты, набравшие определенную сумму баллов или больше, зачисляются на имеющиеся места. Такой балл называется проходным. Если после заполнения имеющихся мест абитуриентами с проходным баллом остаются незаполненные места, но абитуриентов, набравших следующую сумму баллов, больше, чем вакантных мест, набранная этими абитуриентами сумма баллов называется полупро-

ходным баллом. Из числа абитуриентов, набравших полу проходной балл, на имеющиеся места принимаются абитуриенты, имеющие более высокий балл по математике, а при равенстве баллов по математике - по информатике.

Для данного множества абитуриентов следует определить, какая сумма баллов является проходным баллом и какой полу проходной балл по информатике, чтобы быть зачисленным на имеющиеся места.

Входные данные

В первой строке входного файла находятся два числа:  $N$  - количество поданных заявлений о приеме (натуральное число, не превышающее 1000) и  $S$  - количество имеющихся мест. В следующих  $N$  строках три оценки: по русскому языку, математике и информатике, соответственно, разделенные пробелами (все числа натуральные, не превышающие 100).

Запишите в ответе два числа без пробела: сначала проходной балл, затем оценку по информатике, необходимую для зачисления при условии набранного полу проходного балла.

Пример входного файла:

```
42
60 75 90
65 70 90
50 80 100
40 95 80
```

При таких исходных данных проходной балл равен 230, полу проходной 225, на оставшееся одно место будет зачислен абитуриент, набравший в сумме 225 баллов и получивший по информатике 90 баллов.

### Идея решения

---

```
File = open("26.txt")
N, S = map(int, File.readline().split())

sp = []
for i in range(N):
    r, m, i = map(int, File.readline().split())
```

```

# Абитуриент: сумма баллов, математика, информатика, русский
sp.append([m + i + r, m, i, r])

# Сортировка по убыванию
# сначала по сумме баллов, потом по математике, потом по информатике
sp.sort(reverse=True)

ans = [] # Список зачисленных
while len(ans) < S:
    ans.append(sp.pop(0))

# В зависимости от зачисленных абитуриентов
# Определяем ответ на задачу
print(ans)

```

---

## Задача 27

### №27.1

Пусть  $S$  – последовательность из  $N$  чисел пронумерованных подряд начиная с 1. Обозначим  $S_i, S_j, S_k$  три элемента последовательности  $S$ , где  $i < j < k$ .

Определите в последовательности  $S$  три таких числа  $S_i, S_j, S_k$ , что  $S_i > S_j, S_k > S_j$  и значение выражения  $(S_i - S_j) + (S_k - S_j)$  максимально.

В ответе укажите найденное максимальное значение выражения  $(S_i - S_j) + (S_k - S_j)$ .

Гарантируется, что в последовательности есть три числа  $S_i, S_j, S_k$ , удовлетворяющие условию задачи.

Входные данные

Дано два входных файла (файл А и файл В), каждый из которых в первой строке содержит число  $N$  ( $5 < N < 10\,000\,000$ ) – количество целых чисел. Каждая из следующих  $N$  строк содержит одно целое число, значение которого по модулю не превышает 1000.

В ответе укажите два числа: сначала значение искомой величины для файла А, затем - для файла В.

### Решение

---

```
File = open(name)
N = int(File.readline())
a = [int(x) for x in File]
# Список максимальных чисел в порядке слева направо (для числа S_i)
sp_i = [-10**10]*N

curMx = -10**10
for i in range(N):
    curMx = max(curMx, a[i])
    sp_i[i] = curMx

# Список максимальных чисел в порядке справа налево (для числа S_k)
sp_k = [-10**10]*N

curMx = -10 ** 10
for k in range(N-1, -1, -1):
    curMx = max(curMx, a[k])
    sp_k[k] = curMx

res = 0
for j in range(1, N-1): # Перебор числа S_j
    # Для текущего числа S_j:
    # максимальный S_i это sp_i[j-1]
    # максимальный S_k это sp_k[j+1]
    if a[j] < sp_i[j-1] and a[j] < sp_k[j+1]: # Если (S_j < S_i и S_j < S_k)
        res = max(res, (sp_i[j-1] - a[j]) + (sp_k[j+1] - a[j]))
print(res)
```

---

## №27.2 (от 8.06)

Пусть  $S$  - последовательность из  $N$  целых чисел, пронумерованных подряд, начиная с 1. Определите такие индексы  $L, M, R$  ( $0 < L < M < R < N$ ), что разность между суммами чисел с  $L$  по  $M$  и с  $M + 1$  по  $R$  максимальная. В ответе укажите найденное максимальное значение этой разности.

Входные данные:

Дано два входных файла (файл  $A$  и файл  $B$ ), каждый из которых в первой строке содержит число  $N$  ( $5 < N < 1\,000\,000$ ) - количество целых чисел. Каждая из следующих строк содержит одно целое число, значение которого по модулю не превышает 1000. В ответе укажите два числа: сначала значение искомой величины для файла  $A$ , затем - для файла  $B$ .

Пример входного файла:

```
7
-200
-50
500
100
-100
200
50
```

При таких исходных данных искомая величина равна 1000: разность между суммой  $500 + 100 + -100 + 200 + 50$  и суммой  $-200 + -50$ .

В ответе укажите два числа: сначала искомое значение для файла  $A$ , затем для файла  $B$ .

### Решение

```
f = open("27A.txt")
n = int(f.readline())
data = [int(f.readline()) for i in range(n)]
result = float('-inf')
for p in range(2):
```

```

prefix_sum = [0] * (n + 1)
for i in range(1, n + 1):
    prefix_sum[i] = prefix_sum[i - 1] + data[i - 1]

mx_left_sum = [float('-inf')] * (n + 1)
for i in range(n + 1):
    mx_left_sum[i] = max(mx_left_sum[i - 1], prefix_sum[i])

mx_right_sum = [float('-inf')] * n + [prefix_sum[-1]]
for i in range(n - 1, -1, -1):
    mx_right_sum[i] = max(mx_right_sum[i + 1], prefix_sum[i])

for m in range(2, n - 1):
    # Сумма подпосл. от M+1 до R
    sm_r = mx_right_sum[m + 2] - prefix_sum[m]
    # Сумма подпосл. от L до M
    sm_l = prefix_sum[m] - mx_left_sum[m - 2]
    result = max(result, sm_r - sm_l)

data = data[::-1] # Переворачиваем массив
# Это нужно, чтобы сделать то же самое, но наоборот
# То есть (L,M) и (R,M+1) поменяются местами соответственно
# Таким образом, посчитаем оба варианта вычитания:
# sum(M+1:R)-sum(L:M) и sum(L:M)-sum(M+1:R)
print(result)

```

---