

Алгебра логики

$\neg A, \bar{A}$ не А (отрицание, инверсия)
 $A \wedge B, A \cdot B$ А и В (логическое умножение, конъюнкция)
 $A \vee B, A + B$ А или В (логическое сложение, дизъюнкция)
 $A \rightarrow B$ импликация (следование)
 $A \equiv B$ эквивалентность (равносильность)
 $A \rightarrow B = \neg A \vee B$ или $A \rightarrow B = \bar{A} + B$
 формулы де Моргана: $\overline{A \cdot B} = \bar{A} + \bar{B}$ $\overline{A + B} = \bar{A} \cdot \bar{B}$
 $1 + A = 1, 0 + A = 0, A + 0 = A, A \cdot 1 = A, A + A = A$
 $A \cdot A = A, A + \bar{A} = 1, A \cdot \bar{A} = 0, (A + B) \cdot (\bar{A} + B) = B$
 $A \cdot (B + C) = A \cdot B + A \cdot C, A + (B \cdot C) = (A + B) \cdot (A + C)$
 $A + A \cdot B = A, A \cdot (A + B) = A, A \cdot B + \bar{A} \cdot B = B, \bar{\bar{A}} = A$

Задание №2 пример программы (СЛЕДИ ЗА СКОБКАМИ)

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

Задание №4
 Прямое условие Фано: Никакое кодовое слово не может быть началом другого кодового слова.
 Обратное условие Фано: никакой код не был окончанием другого (более длинного) кода.

Внимательно читай, нужно ли использовать весь алфавит! Если нужно, то оставь одно место!

Задание №5

Внимательно читай, что нужно найти!

В двоичной системе:
 - четные числа оканчиваются на 0, нечетные – на 1;
 - числа, которые делятся на 4, оканчиваются на 00, и т.д.; числа, которые делятся на 2^k, оканчиваются на k нулей
 - если число N принадлежит интервалу 2^{k-1} ≤ N < 2^k, в его двоичной записи будет всего k цифр, например, для числа 125:
 2⁶ = 64 ≤ 125 < 128 = 2⁷, 125 = 1111101₂ (7 цифр)
 - числа вида 2^k записываются в двоичной системе как единица и k нулей, например: 16 = 2⁴ = 10000₂
 - числа вида 2^k-1 записываются в двоичной системе k единиц, например: 15 = 2⁴-1 = 1111₂
 - если известна двоичная запись числа N, то двоичную запись числа 2·N можно легко получить, приписав в конец нуля, например: 15 = 1111₂, 30 = 11110₂, 60 = 111100₂, 120 = 1111000₂

Таблица двоичного представления чисел 0-7 в виде триад (групп из 3-х битов):

X ₁₀	X ₈	X ₂
0	000	
1	001	
2	010	
3	011	

X ₁₀	X ₈	X ₂
4	100	
5	101	
6	110	
7	111	

Таблица двоичного представления чисел 0-15 в шестнадцатеричной системе – 0-F₁₆ в виде тетрад (групп из 4-х битов):

X ₁₀	X ₂	X ₁₀	X ₁₆	X ₂
0	0000	8	8	1000
1	0001	9	9	1001
2	0010	10	A	1010
3	0011	11	B	1011
4	0100	12	C	1100
5	0101	13	D	1101
6	0110	14	E	1110
7	0111	15	F	1111

bin(x) представление числа x в двоичной системе

oct(x) представление числа x в восьмеричной системе

hex(x) представление числа x в шестнадцатеричной системе

Задание №6

Если ничего не выводит, то меняй промежуток!

```
for i in range(1, 100):
    s = 1
    n = 1
    while s < 94:
        s = s + 8
        n = n * 2
    if n == 128:
        print(i)
```

Задание: определите, при каком наименьшем введённом значении переменной s программа выведет число 128.

Курсивом выделено, то что было в программе.

Задание №7

1 байт = 8 бит = 2³ бит,
 1 Кбайт = 1024 байта = 2¹⁰ байта = 2¹⁰ · 2³ бит = 2¹³ бит,
 1 Мбайт = 1024 Кбайта = 2¹⁰ Кбайта = 2¹⁰ · 2¹⁰ байта = 2²⁰ байта = 2²⁰ · 2³ бит = 2²³ бит.
 Для хранения растрового изображения нужно выделить в памяти I = x · y · i битов, где x ширина, y высота и i – глубина цвета (разрядность кодирования) Количество цветов = 2ⁱ, i – глубина цвета
 Для хранения информации о звуке длительностью t секунд, закодированном с частотой дискретизации f Гц и глубиной кодирования B бит и количестве каналов k требуется k · B · f · t бит памяти;
 например, при стерео записи (k = 2), f = 8кГц, глубине кодирования 16 бит на отсчёт и длительности звука 128 секунд требуется I = 2 · 8000 · 16 · 128/8/1024/1024 ≈ 3,9 Мбайт

Задание №8

Формула для вычисления числа перестановок с повторениями; для двух разных символов она выглядит так: P(n₁, n₂) = $\frac{(n_1+n_2)!}{n_1! n_2!}$

Здесь n₁ – количество букв А, n₂ – количество звёздочек и восклицательный знак обозначает факториал натурального числа. Число не может начинаться с 0! 0!, 2, 4, 6, 8 четные 1, 3, 5, 7, 9 нечетные

Если задание на СС, то +1 или -1, вспомни про номера слов!

```
n=0
s='школа'
for a in s:
    for b in s:
        for c in s:
            if (a+b+c).count('к')==1:
                n+=1
print(n)
```

Руксами:
 К 4 4 = 16
 4 К 4 = 16
 4 4 К = 16
 Ответ 48

Задание №11

Следить за округлением все вместе или по частям!
 Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля – ровно 11 символов. В качестве символов используются десятичные цифры и 12 различных букв местного алфавита, причём все буквы используются в двух начертаниях: как строчные, так и заглавные (регистр буквы имеет значение!). Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используются посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Определите объём памяти в байтах, который занимает хранение 60 паролей.

Решение:
 - согласно условию, в пароле можно использовать 10 цифр (0..9) + 12 заглавных букв местного алфавита + 12 строчных букв, всего 10 + 12 + 12 = 34 символа
 - для кодирования номера одного из 34 символов нужно выделить 6 бит памяти (5 не хватает, они кодируют только 2⁵ = 32 варианта)
 - для хранения всех 11 символов пароля нужно 11 · 6 = 66 бит – поскольку пароль должен занимать целое число байт, берем ближайшее большее (точнее, не меньшее) значение, которое кратно 8: это 72 = 9 · 8; то есть один пароль занимает 9 байт
 - тогда 60 паролей занимают 9 · 60 = 540 байт **Ответ: 540.**

Задание №12

Если просят найти макс длину, то переставлять числа местами, если можно!
 Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 68 идущих подряд цифр 8? В ответе запишите полученную строку.

```
НАЧАЛО
ПОКА нашлось (222) ИЛИ нашлось (888)
    ЕСЛИ нашлось (222)
        ТО заменить (222, 8)
        ИНАЧЕ заменить (888, 2)
    КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
```

```
n = '8'*68
while '222' in n or '888' in n:
    if '222' in n:
        n = n.replace('222','8',1)
    elif '222' in n:
        n = n.replace('888','2',1)
print(n)
```

Задание №13 Читай вопрос! 5 раз!

Задание №14 Значение арифметического выражения: 125 + 25³ + 5⁹ – записали в системе счисления с основанием 5.

```
#Количество 0 ?
n = 125 + 25**3 + 5**9
k0 = 0
while n > 0:
    if n % 5 == 0:
        k0 += 1
    n = n // 5
print(k0)
```

```
#Для суммы цифр
n = 125 + 25**3 + 5**9
s = 0
while n > 0:
    s += n % 5
    n = n // 5
print(s)
```

Задание №15 Длина отрезка - конец минус начало

Количество точек - конец минус начало + 1
 Читай внимательно, ЦЕЛЫЕ НЕОТРИЦАТЕЛЬНЫЕ ИЛИ ПОЛОЖИТЕЛЬНЫЕ ЗНАЧЕНИЯ!
 ДЕЛ (Для какого наибольшего натурального числа А формула (–ДЕЛ(x, А) ^ ДЕЛ(x, 21)) → ДЕЛ(x, 14) тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x?)

```
for A in range(1,100):
    flag = 0
    for x in range(1,1000):
        if ((x % A != 0) and (x % 21) == 0) <= (x % 14 == 0) == 0:
            flag = 1
    if flag == 0:
        print(A)
```

X и Y

```
for A in range(1,200):
    flag = 0
    for x in range(1,100):
        for y in range(1,100):
            if ((y*y <= A) <= (y <= 10)) and ((x <= 9) <= (x * x < A)) == 0:
                flag = 1
    if flag == 0:
        print(A)
```

Побитовая конъюнкция

```
for A in range(0,100):
    flag = 0
    for x in range(0,1000):
        if ((x & 49 == 0) <= ((x & 28 != 0) <= (x & A != 0))) == 0:
            flag = 1
            break
    if flag == 0:
        print(A)
```

Обозначим через ВЗПР(a, y) утверждение «натуральные числа g и y не имеют общих натуральных делителей, кроме 1».

При каком наименьшем натуральном значении А формула (ВЗПР(x, 360) → ВЗПР(x, А)) ^ (ВЗПР(x, А) → ВЗПР(x, 240)) истинна при любом натуральном x?

```
def gcd1(x, y):
    while y > 0:
        x, y = y, x % y
    if x != 1:
        return False
    else:
        return True
for A in range(2, 100):
    good = 0
    for x in range(1, 1000):
        if (gcd1(x, 360) <= gcd1(x, A)) and (gcd1(x, A) <= gcd1(x, 240)) == False:
            good = 1
            break
    if good == 0:
        print(A)
```

Задание №16

1 функция
 F(n) = n при n ≤ 3;
 F(n) = n // 4 + F(n-3) при 3 < n ≤ 32;
 F(n) = 2 · F(n-5) при n > 32
 Здесь // обозначает деление нацело. В качестве ответа на

2 функции
 Алгоритм вычисления функций F(n) и G(n) задан следующими соотношениями:
 F(1) = G(1) = 1

```
задание выведите значение F(100)
F(n) = 2 * F(n-1) + G(n-1) - 2, если n > 1
G(n) = F(n-1) + 2 * G(n-1), если n > 1
Чему равно значение F(14) + G(14)?
```

```
def f(n):
    if n <= 3:
        return n
    elif n > 3 and n <= 32:
        return n // 4 + f(n-3)
    elif n > 32:
        return 2 * f(n-5)
print(f(100))
```

```
def f(n):
    if n > 1:
        return 2 * f(n-1) + g(n-1) - 2
    else:
        return 1
def g(n):
    if n > 1:
        return f(n-1) + 2 * g(n-1)
    else:
        return 1
print(f(14) + g(14))
```

Задание №17

Обычная (Рассматривается множество целых чисел, принадлежащих отрезку [1012; 9638], которые делятся на 3 и не делятся на 11, 13, 17 и 19. Найдите количество таких чисел и максимальное из них. В ответе запишите два числа через пробел: сначала количество, затем максимальное число.)

```
max1 = 0
k = 0
for i in range(1012, 9638 + 1):
    if (i % 3 == 0) and (i % 11 != 0) and (i % 13 != 0) and (i % 17 != 0) and (i % 19 != 0):
        k += 1
        if i > max1:
            max1 = i
print(k, max1)
```

Сложная (Рассматривается множество целых чисел, принадлежащих числовому отрезку [1000; 9999], запись которых в пятеричной системе имеет не менее 6 цифр и заканчивается на 11 или 13. Найдите количество таких чисел и минимальное из них. ** 6 цифр в пятеричной системе это >5⁵ и <5⁶)

```
count = 0
min_n = 10**8
sum_n = 0
for i in range(1000, 9999 + 1):
    if i >= 3125 and i % 5 == 1 and ((i // 5) % 5 == 1 or (i // 5) % 5 == 3):
        sum_n += i
        count += 1
        if i < min_n:
            min_n = i
print('Количество:', count)
print('Минимальное число:', min_n)
```

Задание №18

Если есть стенки из чисел, то для макс заменяем на 0, для мин заменяем на 999999.
 Перед чем стоит доллар, то и фиксируется. \$A14 фикс столбец А, \$A14, фикс строка 14.

Формула стандартная	B2+МАКС(B12;A13)
Формула для лады (ходит на любое количество клеток по вертикали и горизонтали)	=МАКС(\$Q14;AC14; ADS1:AD13)+N14
Формула, когда с севера на юг	=МАКС(A12:C12)+B2
Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число отличалось от предыдущего не более чем на 10. Какую максимальную сумму могут иметь выбранные числа?	=ЕСЛИ(D1>0;ЕСЛИ(ABS(A2-A1)<=10;D1+A2;A2);A2)

Задание №19-21 Читай каке ходы и когда победа!

Задание №22

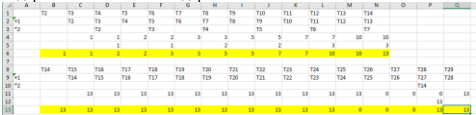
Если ничего не выводит, то меняй промежуток!

```
Получив на вход число x, этот алгоритм печатает два числа: L и M. Укажите наибольшее число x, при вводе которого алгоритм печатает сначала 5, а потом 8.
```

```
for i in range(1, 1000):
    x = 1
    L = 0
    M = 0
    while x > 0:
        M = M + 1
        if x % 2 != 0:
            L = L + 1
        x = x // 2
    if L = 5 and M == 8:
        print(i)
```

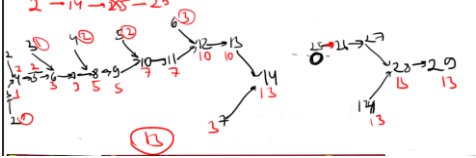
Задание №23

У исполнителя есть две команды, которым присвоены номера: 1. Прибавить 1 2. Умножить на 2
 Сколько существует программ, для которых при исходном числе 2 результатом является число 29 и при этом траектория вычислений содержит число 14 и не содержит числа 25?



```
mas = [0] * 30
mas[2] = 1
for i in range(3, 14 + 1):
    mas[i] += mas[i-1]
    if i % 2 == 0:
        mas[i] += mas[i // 2]
print(mas)
```

```
def f(x, y):
    if x == y:
        return 1
    elif x > y or x == 25:
        return 0
    else:
        return f(x + 1, y) + f(x * 2, y)
print(f(2, 14) * f(14, 29))
```



Задание №24

Следить за краями цикла, если используешь $i + 1, i - 1$.

Одна строка	Несколько строк
<code>f = open("24.txt") s = f.readline() f.close()</code>	<code>f = open("24.txt") for s in f: for i in range(len(s)) f.close()</code>

Функции в питоне

возвращает позицию первой подстроки subs в строке s (или 0 если подстрока не найдена в rascal) -1 в остальных	s.find('subs') S.rfind('l') ищет с конца
заменить в строке S все вхождения подстроки old на подстроку new, count раз	S.replace(old, new, count)
Количество A в строке s	s.count('A')
Получить аски код A	Ord('A')
Превратить аски код в символ	Chr(20)

Пример проги для k = 0

В текстовом файле 24.txt находится цепочка из символов латинского алфавита A, B, C. Найдите длину самой длинной подцепочки, состоящей из символов C.	<code>f = open("24.txt") s = f.readline() f.close() k, kmax = 0, 0 for i in range(len(s)): if s[i] == 'C': k += 1 kmax = max(k, kmax) else: k = 0 print(kmax)</code>
--	--

Пример проги , когда k = 1

Текстовый файл состоит не более чем из 10 ⁶ символов X, Y и Z. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.	<code>f = open("2.txt") s = f.readline() k, maxS = 1, 1 for i in range(1, len(s)): if s[i] != s[i-1]: k += 1 maxS = max(k, maxS) else: k = 1 print(maxS) f.close()</code>
--	---

Пример проги, какая буква встречается чаще всего

Определите символ, который чаще всего встречается в файле сразу после буквы X. В ответе запишите сначала этот символ, а потом сразу (без разделителя) сколько раз он встретился после буквы X.	<code>f = open("1.txt") s = f.readline() a = [0] * 26 nmax, c = 0, 0 for i in range(len(s) - 1): if s[i] == 'X': index = ord(s[i + 1]) - ord('A') a[index] += 1 for i in range(len(a)): if nmax < a[i]: nmax = a[i] c = i print(chr(c + ord('A')), nmax) f.close()</code>
--	--

Текстовый файл состоит не более чем из 10 ⁶ заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Определите количество строк, в которых встречается комбинация F*O, где звездочка обозначает любой символ.	<code>f = open("24.txt") k = 0 for s in f: good = 0 for i in range(len(s)-2): if s[i]=='F' and s[i+2]=='O': good=1 break if good: k+=1 print(k) f.close()</code>
---	--

Задание №25

Все делители числа	Проверка на простоту
<code>del = [] n = int(input()) d = 2 while d * d <= n: if n % d == 0: del.append(d) del.append(n // d) d += 1 if d * d == n: del.append(d) print(del)</code>	<code>def isprime(n): d = 2 while d * d <= n: if n % d == 0: return False d += 1 return True for i in range(2, 100): if isprime(i) == True: print(i)</code>

Решето Эратосфена (нахождение всех простых чисел до n)

<code>n = 20000 resh = [1] * (n + 1) resh[0] = 0 resh[1] = 0 for i in range(2, len(resh)): if resh[i] != 0: for j in range(i * i, len(resh), i): resh[j] = 0 prime = [] for i in range(len(resh)): if resh[i] != 0: prime.append(i) print(prime)</code>

Если нечетное количество делителей, то проверяем только числа, которые являются квадратом другого числа! (n ** 0,5 == int(n ** 0,5))

Алгоритм Евклида для нахождения НОД двух чисел (L и M, L > M)

Через разность	Через остаток
<code>while L != M: if L > M: L -= M else: M -= L print(M) #это НОД</code>	<code>while M > 0: R = L % M L = M M = R Print(L) #это НОД</code>

Задание №26

Сортировка по возрастанию	a.sort()
Сортировка по убыванию	a.sort(reverse = True)
Пузырек	<code>for i in range(len(a)): t = 0 for j in range(len(a) - 1 - i): if a[j] < a[j + 1]: a[j], a[j + 1] = a[j + 1], a[j] t = 1 if t == 0: break</code>

Как копировать в эксель в несколько столбиков :

- 1) Данные – текст по столбцам – с разделителями – далее – пробел – далее – готово. Вставить столбики еще раз
- 2) Файл – открыть – обзор – !!все файлы!! – открыть – тоже самое

Проверять для второго ответа есть ли такие числа в файле!

Задание №27

Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма n всех выбранных чисел не делилась на 3 и при этом была максимально возможной.	<code>f = open("27-B1.txt") n = int(f.readline()) sum1 = 0 div = 100000 for i in range(n): s = f.readline().split() for j in range(len(s)): s[j] = int(s[j]) sum1 += max(s) if max(s) - min(s) < div and (max(s) - min(s)) % 3 != 0: div = max(s) - min(s) if sum1 % 3 != 0: print(sum1) else: print(sum1 - div)</code>
--	--

(ОБЫЧНЫЙ МЕТОД ЧАСТИЧНЫХ СУММ) Набор данных состоит из пар натуральных чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел делилась на 3 и при этом была максимально возможной.	<code>f = open("27-B.txt") mas = [0] * 3 mas1 = [0] * 3 n = int(f.readline()) x = f.readline().split() for j in range(len(x)): x[j] = int(x[j]) mas[min(x) % 3] = min(x) % 3 mas[max(x) % 3] = max(x) % 3 for i in range(1, n): for j in range(len(mas1)): mas1[j] = 0 x = f.readline().split() for j in range(len(x)): x[j] = int(x[j]) for j in range(len(mas)): if mas[j]+x[0]>mas1[(mas[j]+x[0]) % 3]: mas1[(mas[j]+x[0]) % 3] = mas[j]+x[0] if mas[j]+x[1]>mas1[(mas[j]+x[1]) % 3]: mas1[(mas[j]+x[1]) % 3] = mas[j]+x[1] for j in range(len(mas)): mas[j] = mas1[j] print(mas)</code>
---	---

МЕТОД ЧАСТИЧНЫХ СУММ ЧЕРЕЗ ГЕНЕРАТОРЫ	<code>f = open("27-A.txt") n = int(f.readline()) k = 3 mas = list(map(int, f.readline().split())) for i in range(1, n): x = list(map(int, f.readline().split())) gen = [a + b for a in mas for b in x] mas1 = [0] * k for a in gen: mas1[a%k] = max(a, mas1[a%k]) mas = [a for a in mas1 if a != 0] print(mas)</code>
---------------------------------------	---

НЕЭФФЕКТИВН АЯ ПРОГА(СТАРОЕ ЗАДАНИЕ) Все данные – целые числа (возможно, отрицательные). Требуется найти наибольшую сумму двух результатов измерений, выполненных с интервалом не менее, чем в 7 минут.	<code>f = open("27.txt") n = int(f.readline()) a = [0] * n for i in range(n): a[i] = int(f.readline()) maxi = a[0] + a[7] for i in range(n - 7): for j in range(i + 7, n): if a[i] + a[j] > maxi: maxi = a[i] + a[j] print(maxi)</code>
---	--

(СТАРОЕ ЗАДАНИЕ) Дан набор из N натуральных чисел. Необходимо определить количество пар элементов (ai, aj) этого набора, в которых 1 < i < j < N и произведение элементов кратно 6.	<code>f = open("27.txt") n = int(f.readline()) k6, k3, k2, k1 = 0, 0, 0, 0 for i in range(n): a = int(f.readline()) if a % 6 == 0: k6 += 1 elif a % 3 == 0: k3 += 1 elif a % 2 == 0: k2 += 1 else: k1 += 1 print(k6 * (k1 + k2 + k3) + k3 * k2 + k6 * (k6 - 1) // 2)</code>
---	---

(СТАРОЕ ЗАДАНИЕ) Необходимо определить количество пар элементов (ai, aj) этого набора, в которых 1 < i < j < N и сумма элементов кратна 12.	<code>f = open("27.txt") n = int(f.readline()) c = [0] * 12 k = 0 for i in range(n): a = int(f.readline()) c[a % 12] += 1 for i in range(1, 6): k += c[i] * c[12 - i] print(k + c[0] * (c[0] - 1) // 2 + c[6] * (c[6] - 1) // 2)</code>
---	---

(СТАРОЕ ЗАДАНИЕ С ОЧЕРЕДЬЮ) Требуется найти наибольшую сумму двух	<code>f = open("test.txt") n = int(f.readline()) k = 7 buf = [0] * k max_n, maxSum, elem = 0, 0, 0 for i in range(k):</code>
---	--

результатов измерений, выполненных с интервалом не менее, чем в 7 минут.	<code>buf[i] = int(f.readline()) for i in range(k, n): elem = int(f.readline()) if i == k: max_n = buf[0] maxSum = buf[0] + elem else: max_n = max(max_n, buf[0]) maxSum = max(maxSum, max_n + elem) for j in range(k - 1): buf[j] = buf[j + 1] buf[k - 1] = elem print(maxSum)</code>
--	--

(СТАРОЕ ЗАДАНИЕ С ОЧЕРЕДЬЮ)

Необходимо найти в заданной серии количество пар таких показаний прибора, произведение которых кратно 6 и между моментами передачи которых прошло не менее 3 минут.	<code>f = open("27.txt") n = int(f.readline()) k = 3 k6, k3, k2, k1, res = 0, 0, 0, 0, 0 buf = [0] * k for i in range(k): buf[i] = int(f.readline()) for i in range(k, n): if buf[0] % 6 == 0: k6 += 1 elif buf[0] % 3 == 0: k3 += 1 elif buf[0] % 2 == 0: k2 += 1 else: k1 += 1 new1 = int(f.readline()) if new1 % 6 == 0: res += k6 + k3 + k2 + k1 elif new1 % 3 == 0: res += k6 + k2 elif new1 % 2 == 0: res += k6 + k3 else: res += k6 for i in range(k - 1): buf[i] = buf[i + 1] buf[k - 1] = new1 print(res)</code>
---	---