

Всероссийская олимпиада школьников по информатике,
профиль «Программирование»
Вологодская область, 2025-26 учебный год
II (муниципальный) этап,
7 - 8 классы
Методические рекомендации по разбору задач

Задача 1. Игра «Жизнь»

Выполним первые несколько шагов и заметим, что начиная с пятого шага всё начинает повторяться с периодом длины 2. Следовательно, ответ для шага 1000 равен ответу для шага 6, и он равен:

0 0 0 0
0 0 1 0
0 0 1 0
0 0 1 0

Как проделать первые несколько шагов? Во-первых, можно написать программу. Во-вторых, можно их сделать вручную (хотя это немного трудоёмко и легко допустить ошибку). В-третьих, можно воспользоваться электронными таблицами. Например, начало листа в Excel может выглядеть так:

	A	B	C	D	E	F	G	H	I	J
1										
2		0	0	1	1					
3		1	1	0	1					
4		0	0	0	1					
5		1	0	0	0					
6										
7		0	1	1	1		2	3	3	2
8		0	1	0	1		1	2	5	3
9		1	1	1	0		3	3	3	1
10		0	0	0	0		0	1	1	1
11										

Здесь справа вычисляются суммы соседей для таблички слева-сверху. Например, формула в ячейке H8 равна: «=B2+C2+D2+B3+D3+B4+C4+D4»

Затем слева вычисляется новое состояние поля. Например, формула в ячейке C8 равна: «=ЕСЛИ(ИЛИ(И(C3=0;H8=3);И(C3=1;H8>=2;H8<=3));1;0)».

Осталось только выделить диапазон ячеек A6:J10 и скопировать (или протащить мышкой) его ниже, чтобы автоматически получить следующие состояния поля. Дальше несложно заметить наличие периода с пятого шага. Готовый excel-файл прилагается.

Задача 2. Двучётные числа

На первый вопрос можно найти ответ вручную, просто выписав эти числа – их всего 20:

20 22 24 26 28

40 42 44 46 48

60 62 64 66 68

80 82 84 86 88

Далее сделаем такое наблюдение. Рассмотрим двучётные числа с N разрядами, где $N > 2$. Тогда для любой комбинации первых $(N-2)$ цифр существует 25 двучётных чисел, которые с этой комбинации начинаются.

Пояснение. Если сумма цифр в наборе нечётна, то это будут числа вида $xx..x10$, $xx..x12$, $xx..x14$, $xx..x16$, $xx..x18$, $xx..x32$, $xx..34$, ... $xx..96$. Если сумма цифр в наборе чётна, то это будут числа вида $xx..x00$, $xx..x02$, $xx..x04$, $xx..x06$, $xx..x08$, $xx..x22$, $xx..24$, ... $xx..86$.

Теперь ответим на второй вопрос. У нас 4 одноразрядных числа, 20 двухразрядных. Далее 25 трёхразрядных чисел начинаются с цифры 1, ещё 25 – с цифры 2, 25 – с цифры 3, 25 – с цифры 4. Это уже 124 числа. А нам надо 123, то есть, нам нужно предпоследнее трёхзначное двучётное число, начинающееся с цифры 4. Это число 486.

На первые два вопроса также можно было ответить, написав несложную программу, перебирающие натуральные числа по порядку и проверяющие их на двучётность.

Третий вопрос. Возьмём первые 8 разрядов. Из них можно построить $9 \cdot 10^7$ комбинаций, и каждая служит началом для 25 двучётных чисел. Итого $25 \cdot 9 \cdot 10^7 = 2250000000$.

Четвёртый вопрос. Вначале просуммируем все количества с меньшим числом разрядов. $4 + 20 + 225 + 2250 + 22500 + 225000 + 2250000 + 22500000$.

Сколько различных восьмиразрядных чисел можно построить из первых 8 цифр, чтобы числа были меньше 12345678? Это количество равно $12345678 - 10000000 = 2345678$. Для каждого из них есть по 25 окончаний: $25 * 2345678$.

Далее остался префикс 12345678. Заметим, что нам подходят все 25 чисел с таким префиксом. Итого получаем:
 $4 + 20 + 225 + 2250 + 22500 + 225000 + 2250000 + 22500000 + 25 * 2345678 + 25 = 308641974$.

Задача 3. Устричная

За k минут мастер и ученик приготовят $t = a * k + b * (k - 1)$ устриц, назовём это одной порцией устриц. Чтобы приготовить n устриц, нужно приготовить $\lfloor n/t \rfloor$ порций устриц (где $\lfloor \cdot \rfloor$ означает округление вниз), и затем ещё $n \bmod t$ устриц, где $\bmod$ – остаток от деления.

На приготовление $\lfloor n/t \rfloor$ порций уйдёт $\lfloor n/t \rfloor * k$ минут, а на приготовление оставшихся $n \bmod t$ устриц уйдёт $\lceil (n \bmod t) / (a + b) \rceil$ минут (где $\lceil \cdot \rceil$ означает округление вверх). Таким образом, ответ можно вычислить по формуле:

$$\lfloor n/t \rfloor * k + \lceil (n \bmod t) / (a + b) \rceil$$

Чтобы вычислять округление вверх при делении, можно воспользоваться известной формулой $\lceil x/y \rceil = \lfloor (x + y - 1) / y \rfloor = (x + y - 1) // y$, где $//$ – операция деления нацело.

Пример решения на языке Python:

```
a = int(input())
b = int(input())
k = int(input())
n = int(input())
t = a * k + b * (k - 1)
print( (n // t) * k + (n % t + a + b - 1) // (a + b) )
```

Также задача может быть решена при помощи бинарного поиска.

Задача 4. Красивое время

В данной задаче нужно всего лишь аккуратно написать перебор времён в заданном диапазоне (учитывая возможный переход через полночь) и проверку выполнения критериев «красивости».

Пример решения на языке Python:

```
def beautiful(h, m):
    if h == m:
        return True
    if h//10 == m%10 and h%10 == m//10:
        return True
    if h//10+1 == h%10 and h%10+1 == m//10 and m//10+1 == m%10:
        return True
    return False

h1 = int(input())
m1 = int(input())
h2 = int(input())
m2 = int(input())
ans = 0
while True:
    if beautiful(h1, m1):
        ans += 1
    if h1 == h2 and m1 == m2:
        break
    m1 += 1
    if m1 == 60:
        m1 = 0
        h1 += 1
        if h1 == 24:
            h1 = 0
print(ans)
```

Задача 5. Сумма делителей

При $n = 1$ можно вывести любое число и его делитель – оно же само.

При $n = 2$ решения нет.

Случай $n = 3$ дан в условии – ответом будет число 6 с делителями 3, 2 и 1.

Осталось догадаться, что в качестве ответа для числа $n+1$ можно взять удвоенный ответ для числа n . Действительно, пусть m – это ответ для числа n .

Тогда возьмём $2*m$ в качестве ответа, а к списку делителей добавим m . Пример:

$n=4$: $12=6+3+2+1$

$n=5$: $24=12+6+3+2+1$

$n=6$: $48=24+12+6+3+2+1$, и так далее.

Пример решения на языке Python:

```
n = int(input())
if n == 1:
    print(1)
    print(1)
elif n == 2:
```

```
    print(-1)
else:
    x = 3 * 2**(n-2)
    print(x)
    for i in range(n):
        x = (x + 1) // 2
        print(x, end=' ')
```