

**Муниципальный этап Всероссийской олимпиады школьников по информатике по профилю
«Программирование»
2025-2026 учебный год**

Ответы, разбор задач 7-8 класса

Задача А «Доступ к архиву» (100 баллов)

Каждый угаданный код для замка оценивался в 25 баллов.

Способы решения задачи:

- 1) Сопоставление данных о имеющихся цифрах и ручной подбор подходящего числа -
- 2) Программный подбор числа подходящего под перечисленные условия:

Пример программы для получения кода 1 замка:

```
def count_equal_digits(a, b):
    k = 0
    for i in range(len(a)):
        if a[i] == b[i]:
            k += 1
    return k

for i in range(10 ** 4):
    x = f"{i:04d}"
    if (count_equal_digits('1234', x) == 1 and count_equal_digits('2267', x) == 2 and
        count_equal_digits('4589', x) == 0 and count_equal_digits('9030', x) == 2 and
        count_equal_digits('1320', x) == 1):
        print(x)
```

Ответы на задачу:

- 1 – 9260
- 2 – 39542
- 3 – 425341
- 4 – 3282381

Задача В «Упаковка в ящики» (100 баллов)

Ящик имеет прямоугольную форму, поэтому, чтобы найти его площадь, надо знать длины сторон ящика.

Длина стороны складывается из длин ячеек и толщины перегородок.

Размер каждой ячейки равен $2r$, а таких ячеек по одной стороне ящика n штук (суммарная длина получится $2nr$), а по другой стороне – m штук (суммарная длина $2mr$).

Толщина каждой перегородки равна t , а таких перегородок (вместе с внешними бортиками ящика) будет на 1 больше, чем ячеек в ряду, то есть по одной стороне ящика $(n + 1)$ перегородка, а по другой – $(m + 1)$, поэтому они добавят к длинам сторон $(n + 1)t$ и $(m + 1)t$ соответственно. В итоге, первая сторона ящика равна $2nr + (n + 1)t$, а вторая сторона – $2mr + (m + 1)t$.

Для нахождения площади, эти выражения нужно перемножить.

Учитывая требования к записи, итоговый ответ может быть записан в виде:

$$(2 * n * r + (n + 1) * t) * (2 * m * r + (m + 1) * t)$$

Задача С «Космический груз» (100 баллов)

Решение на неполный балл – необходимо последовательно добавлять x или $2 * x$ в зависимости от четности номера дня до тех пор, пока не будет отгружено n модулей. Пример решения на языке Python:

```
n = int(input())
x = int(input())
k = 0
s = 0
while s < n:
    if k % 2 == 0:
        s += x
    else:
        s += 2 * x
    k += 1
print(k)
```

Решение на полный балл:

Вычислим количество пар дней, за которое мы сможем увезти почти все модули $k = n // (3 * x)$. После этого остается вывезти меньше $3 * x$ модулей. Если модулей для вывоза не осталось – ответом является $2 * k$ дней, если модулей осталось $\leq x$, то понадобится $(2 * k + 1)$ день, иначе понадобится еще 1 день.

```
n = int(input())
x = int(input())
k = n // (3 * x)
if n % (3 * x) == 0:
    print(2 * k)
else:
    if n % (3 * x) <= x:
        print(2 * k + 1)
    else:
        print(2 * k + 2)
```

Задача D «Код доступа» (100 баллов)

Для всех значений $r \leq 9$ требуемое количество чисел совпадает с r , поскольку каждое число первого десятка, очевидно, делится на единственную цифру в своей записи. Поэтому числа из оставшегося промежутка $[10; 20]$ легко найти непосредственным проверкой.

Будем перебирать все целые положительные числа n от 1 до r , каждый раз проверяя, делится ли n на свою последнюю цифру. Заметим, что последняя (справа) цифра в десятичной записи заданного числа n совпадает с остатком при делении n на 10, то есть равна $d = n \% 10$.

Последняя цифра, на которую делятся требуемые числа, отлична от нуля, и значит, равна некоторой цифре от 1 до 9. Рассмотрим каждую цифру от 1 до 9 отдельно. Заметим, что все числа, оканчивающиеся на 1 или на 2, делятся на 1 и на 2, то есть удовлетворяют условию задачи. Другими словами, интересными будут все числа вида 1, 11, 21, ..., а также числа вида 2, 12, 22, Сколько таких чисел попадает в промежуток $[1; r]$? Их количество равно $(r - 1)/10 + 1$ — для чисел первого вида, и $(r - 2)/10 + 1$ — для чисел второго вида.

Теперь рассмотрим числа, оканчивающиеся на 3 (или на 6), кратные 3 (или 6) — они образуют арифметическую прогрессию с шагом 30: 3, 33, 63, ... (или 6, 36, 66, ...). Поэтому их количество равно $(r - 3)/30 + 1$ (или $(r - 6)/30 + 1$). Аналогично рассматриваются остальные случаи возможных значений для последних цифр. В каждом из них определяем длину арифметической прогрессии, соответствующей значению последней цифры.

Приведённое решение не требует проверки всех чисел от 1 до r и имеет сложность $O(1)$.

Решение на языке Python:

```
r = int(input())
s1 = (r - 1) // 10 + 1
s2 = (r - 2) // 10 + 1
s3 = (r - 3) // 30 + 1
s4 = (r - 4) // 20 + 1
s5 = (r - 5) // 10 + 1
s6 = (r - 6) // 30 + 1
s7 = (r - 7) // 70 + 1
s8 = (r - 8) // 40 + 1
s9 = (r - 9) // 90 + 1
print(s1 + s2 + s3 + s4 + s5 + s6 + s7 + s8 + s9)
```

Задача E «Галактические шифры» (100 баллов)

Необходимо реализовать алгоритм шифрования, предложенный в условии задачи. Для этого необходимо вычислить количество одинаковых соседних элементов и заменить их последовательность в соответствии с правилами из условия.

Пример решения задачи на языке Python:

```
s = input()

k = 1

arr = []

for i in range(len(s) - 1):

    if s[i] == s[i + 1]:

        k += 1
```

else:

arr.append((s[i], k))

k = 1

arr.append((s[-1], k))

for x in arr:

if x[1] == 1:

print(x[0], end="")

else:

print(x[0], x[1], sep=" ", end="")