

Всероссийская олимпиада школьников по информатике 2025-2026 Профиль “Программирование”

Районный тур, 7-11 класс
Санкт-Петербург, 1 декабря 2025 года

РАЗБОР ЗАДАЧ

Задача А - Ёлочка

— — —

Изобразить ёлочку из
треугольников

```
  *  
 ***  
  *  
 ***  
*****
```

Задача А - решение

- Необходимо вывести треугольники размера от 1 до n
- Чтобы вывести треугольник размера k необходимо вывести k+1 строку
- Пронумеруем строки от 0 до k
- i-я строка получается по формуле

$$\text{"."} \times (n-i) + \text{"*"} \times (2i+1) + \text{"."} \times (n-i)$$

Задача А - код

```
n = int(input())
```

```
s = []
```

```
for i in range(1, n + 1):
```

```
    for j in range(i + 1):
```

```
        s.append('.'*(n - j) + '*'*(2 * j + 1) + '.'*(n - j))
```

```
print("\n".join(s))
```

Задача В - Лифт

— — —

В лифте кнопки расположены в два столбца

Маленький мальчик достаёт только до нижних s кнопок

Хочет приехать на k -й этаж, но возможно часть пути придется пройти по лестнице

n	$2n$
$n-1$	$2n-1$
...	...
s	$n+s$
$s-1$	$n+s-1$
...	...
1	$n+1$

Задача В - Лифт - посмотрим, где расположен этаж k

— — —

n $2n$

$n-1$ $2n-1$

... ...

s $n+s$

$s-1$ $n+s-1$

... ...

1 $n+1$

← k

Если он между 1 и s
или между $n+1$ и $n+s$
(включительно) – можно
доехать до самого
этажа

Задача В - Лифт - посмотрим, где расположен этаж k

— — —

n $2n$

$n-1$ $2n-1$

...

← k

s $n+s$

$s-1$ $n+s-1$

...

1 $n+1$

Если он между $s+1$ и n
или между $n+s+1$ и $2n$,
то можно подняться от
 s -го, подняться от
 $n+s$ -го или спуститься
с $n+1$ -го

Задача В - Особый случай

Если $s = 0$, то мальчик вообще не достает до кнопок

n	$2n$
$n-1$	$2n-1$
...	...

s	$n+s$
$s-1$	$n+s-1$
...	...
1	$n+1$

Задача В - Код

```
n = int(input())
k = int(input())
s = int(input())

if (s == 0):
    print(k - 1)
elif k <= s or (k >= n + 1 and k <= n + s):
    print(0)
elif k <= n:
    print(min(k - s, n + 1 - k))
else:
    print((k - n) - s)
```

Задача С - Сортировка комнат

— — —

Комната имеет размер $a_i \times b_i$

В ней выделен вентиляционный короб размера $w_i \times h_i$

Необходимо отсортировать комнаты по площади

Задача С - Сортировка комнат

— — —

Комната имеет размер $a_i \times b_i$ метров

В ней выделен вентиляционный короб размера $w_i \times h_i$ см

Необходимо отсортировать комнаты по площади

Задача С - Сортировка комнат - формула

— — —

Комната имеет размер $a_i \times b_i$ метров

В ней выделен вентиляционный короб размера $w_i \times h_i$ см

Необходимо отсортировать комнаты по площади

Площадь комнаты в см^2

$$S[i] = a[i] * b[i] * 10000 - h[i] * w[i]$$

Используем встроенную сортировку

Задача C - код

— — —

```
n = int(input())
a = []

for i in range(n):
    w, h, x, y = map(int, input().split())
    w *= 100
    h *= 100
    s = (w * h - x * y, i + 1)
    a.append(s)

a.sort()
for room in a:
    print(room[1], end = " ")
```

Задача D - разнообразные строки

Строка называется разнообразной, если в ней есть хотя бы две различные буквы

Найти число подстрок, которые являются разнообразными

Подстрока задается индексами

Пример: “cool”

Ответ: 5 (“co”, “coo”, “cool”, “ool”, “ol”)

Задача D - решение

Давайте считать “скучные” строки, в которых все символы одинаковые

Если подряд идет k одинаковых символов, то это дает нам $k*(k+1)/2$ скучных подстрок

А всего подстрок $n*(n+1)/2$

Задача D - код

— — —

```
s = input()
n = len(s)
s += "$"

same = 1
boring = 0

for i in range(1, n + 1):
    if s[i] != s[i - 1]:
        boring += same * (same + 1) // 2
        same = 1
    else:
        same += 1

print(n * (n + 1) // 2 - boring)
```

Задача E - сумма цифр

— — —

Цифровой суммой разбиения на слагаемые называется сумма цифр слагаемых

Нужно найти разбиения n с заданной цифровой суммой

12 3

1 1 10

1 11

2 10

12

Задача E - решение

— — —

Заметим, что сумма цифр по модулю 9 сохраняется при суммировании чисел (это так называемый “цифровой корень”

Поэтому если сумма цифр n и s по модулю 9 различаются, ответ равен 0

Иначе можно делать перебор с отсечениями

Задача E - функция для суммы цифр

```
int sum_digits(int n) {  
    int r = 0;  
    while (n > 0) {  
        r += n % 10;  
        n /= 10;  
    }  
    return r;  
}
```

Задача E - рекурсивный перебор

— — —

```
vector<vector<int>> ans;
void bt(int p, int last, int n, int s) {
    if (n == 0 && s == 0) {
        ans.push_back(a);
        return;
    }
    if (n < s) return;
    if (s == 0) return;

    for (int i = last; i <= n; i++) {
        int d = sum_digits(i);

        if (d <= s) {
            a.push_back(i);
            bt(p + 1, i, n - i, s - d);
            a.pop_back();
        }
    }
}
```

Задача F - подарки

Необходимо найти такие n и k , чтобы число сочетаний из n по k равнялось заданному числу z

При прочих равных минимизировать n

Число z может быть огромным, до 10^{100}

Задача F - первая подзадача $z \leq 10$, ручной разбор

$z = 1: n = 1$

1

$z = 2: n = 2$

1 2 1

$z = 3: n = 3$

1 3 3 1

$z = 4: n = 4$

1 4 6 4 1

$z = 5: n = 5$

1 5 10 10 5 1

$z = 6: \mathbf{n = 4}$

$z = 7: n = 7$

$z = 8: n = 8$

$z = 9: n = 9$

$z = 10: \mathbf{n = 5}$

Задача F - вторая подзадача $z \leq 1000$

Ясно, что $n \leq z$

Построим треугольник Паскаля для $n \leq 1000$

Поищем наше значение

Осторожно, может случиться переполнение

- на Python все ок
- на C++ можно заменить все числа ≥ 1000 на 1001

Задача F - третья подзадача, $z \leq 10^9$

Заметим, что k не слишком большое

Посмотрим на формулу

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

Преобразуем её так:

$$C_n^k = \frac{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1)}{1 \cdot 2 \cdot \dots \cdot k}$$

Задача F - решение

$$C_n^k = \frac{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1)}{1 \cdot 2 \cdot \dots \cdot k}$$

Видно, что если k мало по сравнению с n , то формула ведет себя почти как n^k

Будем перебирать k по возрастанию, для данного k подбирать n

Значение n можно подобрать бинарным поиском, либо линейным поиском (кроме $k = 1$, но тогда $n = z$)

Задача F - решение

- Когда перестать перебирать k ?
- Если k такое большое, что $C(2k, k) > z$, то можно остановиться, для всех $n \geq 2k$ будет также $C(n, k) > z$, а для $n < 2k$ подходит также меньшее $k' = n - k$

$$C_n^{n-k} = C_n^k$$