

# Отборочный тур

## Много ли чисел

**Ответ:** 45. **Решение.** Попробуем перебрать такие числа. С единицей в числе десятков такое число одно, с двойкой — два, с тройкой — три, ..., с девяткой — девять. Сумма цифр от одного до девяти равна 45.

## Интересное число

**Ответ:** 5999995. **Решение.** Для того, чтобы число было наименьшим, нам нужно использовать как можно меньше цифр. Чтобы набрать сумму цифр 55 как можно меньшим количеством цифр, нужно использовать как можно большие цифры. Девятками мы можем набрать максимум 45. Чтобы число было кратно 5, нужно чтобы оно оканчивалось на 0 или 5. С нулем на конце наименьшим будет 19999990, а с пятеркой на конце 5999995. Меньшее из этих двух чисел и является ответом.

## 2018-ый член последовательности

**Ответ:** 4074342. **Решение.** Заметим, что  $n$ -ый член последовательности находится по формуле  $n \cdot (n + 1)$ . Следовательно, 2018-ый член последовательности равен  $2018 \cdot 2019 = 4074342$ .

## Угадывание числа

**Ответ:** 9. **Решение.** Воспользуемся методом двоичного поиска. Каждым вопросом мы можем отсекаем половину чисел. Для этого первым вопросом будет: “Больше ли загаданное число, чем  $365 // 2 = 182$  ?” ( $//$  -- целочисленное деление). Затем благодаря ответу мы определим в каком промежутке находится загаданное число: от 1 до 182 или от 183 до 365. Полученный промежуток поделит пополам и зададим аналогичный вопрос и так далее. Всего нам потребуется 9 вопросов, так как  $2^9 = 512 > 365$ .

## Шарик и Матроскин

**Ответ:** 50. **Решение.** Для каждого двузначного числа  $n$  его парой, дающей в сумме 100 является число  $100 - n$ . Однозначные числа от 1 до 9 составляют пары числам от 99 до 91, следовательно их точно можно написать. Из оставшегося 81 двузначного числа можно максимум выписать 41. Проиллюстрируем это: все числа, кроме 50 можно разбить на пары, дающие в сумме 100. Из каждой такой пары можем выбрать не более одного числа. Следовательно, всего можно выписать максимум 50 чисел.

## Миньоны

**Ответ:** 178. **Решение.** Между 90 Миньонами 89 интервалов. Так как каждый из них равен 2 метрам, то всего Грю пройдет  $89 \cdot 2 = 178$ .

## Кашей и Василиса

**Ответ:** 6. **Решение.** Назовем  $T_n$  -- число монет, из которых Кашей точно сможет определить фальшивую, затратив при этом не больше  $n$  золотых. В таком случае, на каждую из чаш можно положить не более  $T_{n-1}$  монет (так как в случае неравенства,

Кашею нужно будет заплатить 1 золотой, а затем определять фальшивую монету из  $T_{n-1}$  монет), а вне чашек может остаться соответственно не больше  $T_{n-2}$  монет.

Значит, можем составить такое неравенство:  $T_n \leq T_{n-2} + 2T_{n-1}$

Очевидно, что  $T_0 = 1$  (если ничего не взвешивать, то фальшивая монета должна быть единственной),  $T_1 = 2$ . Затем, можем последовательно найти, что

$T_2 \leq 5$ ,  $T_3 \leq 12$ ,  $T_4 \leq 29$ ,  $T_5 \leq 70$ ,  $T_6 \leq 169$

Покажем, что 6 золотых нам хватит. Положим на две чаши весов по 70 монет и 29 монет отложим. Тогда если весы окажутся в равновесии, то за 4 оставшихся золотых, мы сможем найти фальшивую монету среди 29 монет, и т.д., а если одна из чаш перевесит, то за 5 оставшихся золотых мы сможем найти фальшивую монету среди 70 монет и т.д.

## Аня ищет числа

**Ответ:** 15. **Решение.** Запишем условие как:  $7 \cdot (10a + b) = 100a + b$ . Получим:

$$70a + 7b = 100a + b$$

$$6b = 30a$$

$$b = 5a$$

А так как  $a$  и  $b$  - цифры, то  $b = 5$ ,  $a = 1$ .

## Чистка зубов

В задаче нужно было проверить, что число во вводе четное. Для этого нужно взять остаток от деления этого числа на 2 и проверить, что он равен нулю.

Пример решения на Python:

```
n = int(input())
if n % 2 == 0:
    print("Yes")
else:
    print("No")
```

## Дверь для Кроша

В этой задаче нужно было проверить, что круг можно поместить в прямоугольник. Для этого нужно проверить, что диаметр круга не больше каждой из сторон прямоугольника.

Пример решения на Python:

```
a, b, r = map(int, input().split())
if r <= min(a, b):
    print("Yes")
else:
    print("No")
```

## Конфетки

Сначала найдем, сколько коробок конфет нужно купить. Для этого нужно разделить  $n$  на  $c$ , округлив ответ вверх. Это можно сделать разными способами, проще всего

использовать формулу  $(n + c - 1) // c$  (где `//` — деление с округлением вниз).  
Далее умножим полученное число коробок на цену одной коробки.

Пример решения на Python:

```
n, c, p = map(int, input().split())
print((n + c - 1) // c * p)
```

## Коллекция покемонов

В этой задаче нужно было найти число различных элементов в массиве.

Есть много способов решить эту задачу. Например, можно отсортировать массив, тогда одинаковые элементы будут рядом. Теперь посчитаем только те элементы, которые не равны стоящему слева от них.

```
n = int(input())
a = list(map(int, input().split()))
a.sort()
c = 0
for i in range(n):
    if i == 0 or a[i - 1] != a[i]:
        c += 1
print(c)
```

Другой способ — использовать структуру данных `set` для хранения множества чисел, которые есть в массиве

```
n = int(input())
a = list(map(int, input().split()))
c = set()
for i in range(n):
    c.add(a[i])
print(len(c))
```

## Секретное сообщение

В этой задаче нужно взять первую букву из каждого слова в сообщении и соединить их в одно слово. Для того, чтобы разделить ввод на отдельные слова, в Python можно использовать операцию `split()`. Далее нужно пройтись по всем словам и выписать их первые буквы.

```
a = input().split()
for s in a:
    print(s[0], end="")
```

## Будильник

В этой задаче нужно научиться вычислять, какое время будет через 9 минут. Для этого увеличим текущее число минут на 9, если оно стало больше или равно 60, то увеличим число часов, если оно стало больше или равно 24, то перейдем на следующие сутки.

Пример решения на Python:

```
h, m = map(int, input().split())
for i in range(10):
    print(h, m)
    m += 9
    if m >= 60:
        h += 1
        m -= 60
    if h >= 24:
        h -= 24
```

## Акция

Посмотрим, какую самую дорогую книгу мы можем получить бесплатно по акции. Это третья по цене книга. Для того, чтобы получить ее бесплатно, нужно купить группу из трех самых дорогих книг. Продолжая рассуждения тем же способом, получим, что далее, чтобы получить наибольшую скидку, нужно купить три следующие по цене книги, и т. д. Таким образом, получаем следующий алгоритм.

Отсортируем книги по убыванию цены. Формируем группы (1, 2, 3), (4, 5, 6), (7, 8, 9), ... пока книг хотя бы три. В каждой группе третья книга бесплатная.

Пример решения на Python:

```
n = int(input())
a = [int(input()) for i in range(n)]
a.sort()
a = a[::-1]
s = 0
for i in range(n):
    if i % 3 != 2:
        s += a[i]
print(s)
```

## Кенгуренок

Для того, чтобы число прыжков было минимальным, нужно делать как можно меньше прыжков назад. Если число  $d$  делится на 3, то можно не делать ни одного прыжка назад. Если остаток от деления на 3 равен 1, то нужно сделать один прыжок назад, если остаток равен 2, то нужно два прыжка назад. Чтобы упростить код, сделаем цикл **while**, который делает прыжки назад, пока расстояние до нужной точки не будет делиться на 3.

Пример решения на Python:

```
d = int(input())
ans = []
while d % 3 != 0:
    ans.append(-2)
    d += 2
while d > 0:
    ans.append(3)
    d -= 3
print(len(ans))
print(*ans)
```

## Разворот отрезков

Есть несколько способов решить эту задачу. Приведем не самый эффективный, зато очень простой. Будем каждый раз находить самые левые открывающую и закрывающую скобки и разворачивать подстроку между ними.

Пример решения на Python:

```
s = input()
while True:
    l = s.find("(")
    r = s.find(")")
    if l == -1:
        break
    s = s[:l] + s[l + 1: r][::-1] + s[r + 1:]
print(s)
```

## Веревочная почта

Пусть  $D$  — текущий сдвиг веревки от начального положения. Чтобы человек  $i$  мог передать сообщение человеку  $a[i]$ , веревка должна быть сдвинута на  $(a[i] - i)$ . Таким образом, задача сводится к следующей. Есть число  $D$ , изначально равное нулю. За одно действие можно увеличить или уменьшить его на 1. Нужно, чтобы это число приняло каждое из значений  $(a[i] - i)$  для всех заданных  $i$ . Легко заметить, что для того, чтобы это условие было выполнено, необходимо и достаточно, чтобы число  $D$  добежало до минимального и максимального из чисел  $(a[i] - i)$ . Таким образом есть два варианта движения: дойти до минимального  $(a[i] - i)$ , затем до максимального, или наоборот. Выберем из двух этих вариантов минимальный.

Пример решения на Python:

```
n = int(input())
a = list(map(int, input().split()))
minD, maxD = 0, 0
for i in range(n):
    minD = min(minD, a[i] - (i + 1))
```

```
maxD = max(maxD, a[i] - (i + 1))  
print(min(-minD * 2 + maxD, -minD + maxD * 2))
```

# Финал

## Слоны на водопое

**Ответ:** 21. **Решение.** Остались только слоны пришедшие позавчера, вчера и сегодня.  $7 \cdot 3 = 21$ .

## Королевский садовник

**Ответ:** 60. **Решение.** На третий день появилась клумба с  $(80+40)/2=60$  цветами. И каждый следующий день в новой клумбе было так же 60 цветов, так как если в группу чисел добавить число, равное среднему арифметическому этой группы, то среднее арифметическое новой группы будет равно среднему арифметическому начальной группы.

## Изменение чисел

**Ответ:** 2018. **Решение.**  $2018 = 2099 - 81$  Именно 2099 даст через 81 минуту наибольшее число, так как любое число большее его даст в результате трехзначное число.

## Гири от 1 до 13

**Ответ:** 1 3 9. **Решение.** При решении этой задачи нам понадобится следующее интересное свойство троичной системы счисления: любое натуральное число можно представить в виде разности двух чисел, запись которых в троичной системе счисления содержит только 0 и 1.

Для доказательства нужно записать исходное число в троичной системе счисления и построить требуемые числа поразрядно справа налево. При этом если у получившихся чисел в каких-то одноименных разрядах стоят единицы, то их можно заменить нулями.

Теперь понятно, что нам нужны 3 гири с весами 1, 3 и 9. Двух гирь недостаточно, так как с их помощью можно взвесить максимум  $3^2 - 1 = 8$  различных весов.

## Гири от 1 до 121

**Ответ:** 1 3 9 27 81. **Решение.** При решении этой задачи нам понадобится следующее интересное свойство троичной системы счисления: любое натуральное число можно представить в виде разности двух чисел, запись которых в троичной системе счисления содержит только 0 и 1.

Для доказательства нужно записать исходное число в троичной системе счисления и построить требуемые числа поразрядно справа налево. При этом если у получившихся чисел в каких-то одноименных разрядах стоят единицы, то их можно заменить нулями.

Теперь понятно, что нам нужны 5 гирь с весами 1, 3, 9, 27 и 81. Двух гирь недостаточно, так как с их помощью можно взвесить максимум  $3^4 - 1 = 80$  различных весов.

## Моана

**Ответ:** 1500. **Решение.** Нам нужны числа-палиндромы, следовательно достаточно подсчитать количество различных способов выбрать первые 5 цифр и остальные расставятся автоматически. На первое место мы можем поставить любую из 4 цифр: 1, 8, 6, 9. На второе -- четвертое место мы можем поставить любую из 5 цифр: 1, 8, 6, 9, 0. На пятое место мы не можем поставить 6 и 9, следовательно остается 3 способа выбрать пятую цифру. Итого, получаем:  $4 \cdot 5 \cdot 5 \cdot 5 \cdot 3 = 1500$  чисел.

## Отличник Поликарп

**Ответ:** 3. **Решение.** Из 500 цифр, стёртых Колькой, на однозначные числа уйдёт 9 цифр, значит, на остальные останется 491 цифра. На двузначные числа уйдёт  $90 \cdot 2 = 180$  цифр, значит, на остальные останется 311 цифр. Из этого количества цифр получится 103 трехзначных числа и ещё две цифры от 104-го. Это значит, что интересующая нас цифра — 3-я цифра 104-го трехзначного числа. Это число 203, значит, искомая цифра 3.

## Шоколадка

Прямоугольную шоколадку можно разделить на две равные части тогда и только тогда, когда одна из сторон четная.

Пример решения на Python:

```
n, m = map(int, input().split())
if n % 2 == 0 or m % 2 == 0:
    print("Yes")
else:
    print("No")
```

## Тортики

Есть несколько вариантов решения. Например, можно просто печь торты по одному и хранить текущее число ингредиентов.

```
a, b, c, x, y, z = map(int, input().split())
ans = 0
while x >= a and y >= b and z >= c:
    x -= a
    y -= b
    z -= c
    ans += 1
print(ans)
```

Более эффективное решение — посчитать для каждого ингредиента, на сколько тортов его хватит, и взять минимум из этих значений.

```
a, b, c, x, y, z = map(int, input().split())
print(min(x // a, y // b, z // c))
```



## Кольцевая

Есть разные способы решать эту задачу. Основная идея в том, чтобы проверить, в какую сторону ехать быстрее — по часовой стрелке или против. Например, можно использовать такие формулы: если ехать по часовой стрелке, то нужно проехать  $(b - a) \% n$  станций, а если против часовой —  $(a - b) \% n$  станций.

Пример решения на Python:

```
n, a, b = map(int, input().split())
print(min((b - a) % n, (a - b) % n))
```

## Машинка

Будем выбирать пары одинаковых пуговиц, пока они есть. Если пар найдется хотя бы две, то выведем ответ, иначе ответа нет. Для того, чтобы быстро находить пары одинаковых пуговиц, можно отсортировать массив, тогда одинаковые элементы массива будут рядом.

Пример решения на Python:

```
n = int(input())
a = list(map(int, input().split()))
a.sort()
ans = []
i = 0
while i < n - 1:
    if a[i] == a[i + 1]:
        ans.append(a[i])
        i += 2
    else:
        i += 1
if len(ans) < 2:
    print(-1)
else:
    print(ans[0], ans[1])
```

## Матрешки

Есть множество способов решить эту задачу. Например, можно взять отсортированный массив и затем поменять четные элементы с предыдущими нечетными.

Пример решения на Python:

```
n = int(input())
a = [(i + 1) for i in range(n)]
for i in range(1, n, 2):
    a[i - 1], a[i] = a[i], a[i - 1]
```

```
print(*a)
```

## Массив

Всего существует  $(B - A + 1)$  чисел в диапазоне от  $A$  до  $B$ . Если  $n$  больше, чем это число, то ответа нет. Иначе построим массив, взяв числа  $A, B$ , и любые  $(n - 2)$  чисел из диапазона от  $A + 1$  до  $B - 1$ .

Пример решения на Python:

```
n, A, B = map(int, input().split())
if n > (B - A + 1):
    print(-1)
else:
    ans = [A, B] + [A + i for i in range(1, n - 1)]
    print(*ans)
```

## Змей Горыныч

Одно из решений такое: сначала отрубаем головы по одной, остаются только хвосты. Потом отрубаем по одному хвосту, пока число хвостов не будет делиться на 4. Далее отрубаем по два хвоста, вырастает четное число голов. Отрубаем их по две.

Пример решения на Python:

```
A, B = map(int, input().split())
while A > 0:
    print("H")
    A -= 1
    B += 1
while B % 4 != 0:
    print("T")
    B += 1
while B > 0:
    print("TT")
    B -= 2
    A += 1
while A > 0:
    print("HH")
    A -= 2
```

## Сумма цифр

Пусть номер первой страницы на развороте  $x$ , а второй —  $x + 1$ . Поскольку число  $x$  четное, число  $x + 1$  отличается от него только последней цифрой. Таким образом, сумма цифр числа  $x + 1$  на один больше, чем сумма цифр числа  $x$ .

$$S = \text{sum}(X) + \text{sum}(X + 1) = 2 \cdot \text{sum}(X) + 1$$

Таким образом,  $\text{sum}(x) = (s - 1) / 2$ . Теперь задача свелась к следующей: найти минимальное четное число  $x$  с заданной суммой цифр. Если сумма хотя бы двузначная, то число всегда выглядит так:  $x999\dots9998$ , однозначные суммы разберем отдельным случаем.

Пример решения на Python:

```
s = int(input())
if s == 1:
    print(1)
elif s % 2 == 0:
    print(-1)
else:
    s //= 2
    if s >= 10:
        x = "8"
        s -= 8
        while s > 9:
            s -= 9
            x = "9" + x
        x = str(s) + x
        print(x, int(x) + 1)
    else:
        if s % 2 == 1:
            s = 10 + s - 1
        print(s, s + 1)
```