Задача 1. Предыдущий палиндром

Ответ:
12321
98689
219912
899998
999999

Задача 2. Ремонт площади

К каждому отрезку длиной 1 м каждой боковой стороны прилегает ровно одна треугольная плитка. Сумма длин всех боковых сторон равна $2n + 2m$. При этом угловые плитки (их четыре) посчитали по два раза каждую, поскольку они прилегают к двум боковым сторонам. Из этого выражения нужно вычесть 4.

Ответ (допускаются и другие формы записи ответа): $2 * n + 2 * m - 4$

Задача 3. Алгоритм переливаний

Несложно заметить, что используя только первые две колбы нельзя отмерить 1 мл, потому что объёмы первой и второй колбы делятся на 2, поэтому мы всегда будем получать чётное число мл реактива. Поэтому обязательно использовать третью колбу. Но если общий объём использованного реактива будет меньше 35 (меньше объёма третьей колбы), то это будет означать, что мы по сути не использовали тот факт, что объём третьей колбы равен 35: если мы заменим эту колбу на колбу любого большего размера (например, на колбу размером 36) и повторим все действия, то получим тот же самый результат, и в этом случае объёмы воды также всегда будут кратны 2 мл. То есть нельзя придумать решение, использующее менее 35 мл реактива.

Пример решения, использующего ровно 35 мл реактива:

```
>C
C>B
B>A
B>
A>B
C>B
```

Задача 4. Музей

Попробуем построить наилучший ответ. Заметим, что не имеет смысла делать стрелки, направленные вне музея — всегда можно заменить такую стрелку на стрелку внутрь музея, и ответ не станет хуже. Тогда маршрут каждого человека представляет некоторую ациклическую часть, после чего он будет перемещаться по какому-то циклу, и множества людей разобьются на группы, двигающиеся по циклам. Чтобы каждый человек посетил как можно больше залов, было бы желательно объединить все залы в один цикл. Однако это сделать нельзя, потому что общее число залов — нечётное, а длина любого цикла будет чётной (это легко доказать, рассмотрев шахматную раскраску залов). Но можно объединить 24 зала в один цикл. Посетители, находившиеся первоначально в этих залах, посетят по 24 зала. Из оставшегося зала, не включенного в цикл, направим стрелку в любой зал цикла, тогда посетитель этого зала побывает во всех 25 залах (своём и всех залах цикла).

Пример такого решение, где цикл охватывает все залы, кроме левого нижнего угла:

```
RRRRD
UDLLL
URRRD
```

```
ULDLD
UULUL
```

Задача 5. Пятистенок

Для строительства одного ряда нужно три бревна длиной a и два бревна длиной b, поэтому для нахождения ответа нужно нацело поделить n на $3a + 2b$.

```
a = int(input())
b = int(input())
c = int(input())
ans = c // (3 * a + 2 * b)
print(ans)
```

Задача 6. Ёлочки

Ёлочка красоты n состоит из $2n$ ветвей длины 1, 2, ..., n и ствола длины $2n + 1$, поэтому нужно посчитать сумму $s = 1 + 2 + ... + n$ и вывести значение $2s + 2n + 1$.

Если для нахождения суммы чисел от 1 до n использовать цикл, то решение будет набирать 60 баллов. т.к. для больших n будет работать долго. Пример такого решения

```
n = int(input())
s = 0
for i in range(1, n + 1):
    s += i
print(2 * s + 2 * n + 1)
```

Для того, чтобы набрать 100 баллов, заметим, что $2s = n(n + 1)$, поэтому программа может просто вывести значение $n(n + 1) + 2n + 1$.

```
n = int(input())
print(n * (n + 1) + 2 * n + 1)
```

Задача 7. Соревнование делимости

Для того, чтобы набрать 60 баллов, можно просто перебрать все числа от x до y, посчитать число делящихся на k среди них, посчитать число делящихся на m, и вычесть из одного числа другое. В решении ниже эти количества не считаются отдельно, а считается сразу же ответ. Если число делится на k, то ответ увеличивается на 1, если число делится на m, то ответ уменьшается на 1, одно число может и увеличить, и сразу же уменьшить ответ на 1.

```
k = int(input())
m = int(input())
x = int(input())
y = int(input())

ans = 0
for i in range(x, y + 1):
    if i % k == 0:
        ans += 1
    if i % m == 0:
        ans -= 1
print(ans)
```

Чтобы набрать 100 баллов, нужно быстро посчитать количество чисел, делящихся на данное число на данном отрезке. Количество чисел от 1 до y, делящихся на k, равно $\lfloor \frac{y}{k} \rfloor$ (целая часть от частного $\frac{y}{k}$, то есть целочисленное деление y на k). А чтобы посчитать количество кратных k на отрезке от x до y нужно из количества кратных k на отрезке от 1 до y вычесть количество кратных k на отрезке от 1 до $x - 1$, то есть $\lfloor \frac{y}{k} \rfloor - \lfloor \frac{x-1}{k} \rfloor$. Аналогично посчитаем количество кратных m на отрезке от x до y. Пример решения.

```
k = int(input())
m = int(input())
x = int(input())
y = int(input())
print(y // k - (x - 1) // k - y // m + (x - 1) // m)
```