

Задача А. Скидка на телефоны

Чтобы купить оба телефона без сдачи, необходимо взять с собой сумму, равную общей стоимости товаров. На товары действует скидка c , поэтому первый телефон стоит $a - c$, второй $b - c$. Следовательно, нужно заплатить $(a - c) + (b - c) = a + b - 2c$.

```
a = int(input())
b = int(input())
c = int(input())
print(a + b - 2 * c)
```

Задача В. Лучшая работа

В данной задаче достаточно посчитать, сколько Вова будет зарабатывать в каждом из случаев. В первом варианте Вова будет зарабатывать $a \cdot b$ рублей за день, а во втором варианте $c \cdot d$ рублей. Максимум из этих двух величин и будет являться ответом.

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())

print(max(a * b, c * d))
```

Задача С. Урок литературы

Посчитаем количество людей, которые прочитали перед нами, тогда это число, увеличенное на 1, будет ответом. Если сидим в ряду a на b -й парте и на варианте c , то перед нами прочитали $a - 1$ рядов, $b - 1$ парт в нашем ряду и $c - 1$ учеников, которые сидят за нашей партой. В каждом ряду $2 \cdot n$ людей, за каждой партой два человека, значит перед нами прочитало $2 \cdot (a - 1) \cdot n + 2 \cdot (b - 1) + c - 1$. Осталось это число увеличить на 1, так как нужно вывести номер предложения, которое читаем.

```
n = int(input())
a = int(input())
b = int(input())
c = int(input())
print(2 * (a - 1) * n + 2 * (b - 1) + c)
```

Задача D. Задача Архимуса

Чтобы решить данную задачу, можно было рассмотреть все варианты разбиения палочек на пары. Аналогично можно было проверить только разбиение, в котором первая пара содержит меньшие палочки, а вторая большие.

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())

sticks = [a, b, c, d]
sticks.sort()

if sticks[0] == sticks[1] and sticks[2] == sticks[3]:
    print("Yes")
else:
    print("No")
```

Задача Е. Припев песни

Так как припев длится l секунд, начинается на a -й и b -й секунду, то первый раз он играет на отрезке $[a : a + l - 1]$, второй раз на отрезке $[b : b + l - 1]$. Разберем 5 случаев:

1) $c < a$, то есть первый припев ещё не начался, тогда должны дождаться его начала, значит ответ будет $a - c$.

2) $a \leq c \leq a + l - 1$, то есть сейчас Петя слушает припев, тогда ответ 0.

3) $a + l \leq c < b$, первый припев уже закончился, а второй ещё не начался, тогда должны дождаться его начала и ответ будет $b - c$.

4) $b \leq c \leq b + l - 1$, сейчас Петя слушает припев, тогда ответ 0.

5) $b + l \leq c$, второй припев уже закончился, так как песня зациклена, то нужно дождаться первого припева и ответ $t - c + a$.

```
t = int(input())
a = int(input())
b = int(input())
l = int(input())
c = int(input())

if a <= c < a + l or b <= c < b + l:
    print(0)
elif a + l <= c < b:
    print(b - c)
elif c < a:
    print(a - c)
else:
    print(t - c + a)
```

Задача F. Уменьшить строку

Сделаем следующий алгоритм : сожмем все подряд идущие символы в один. Если остался один символ, то дальше уменьшить строку нельзя и ответ 0. В ином случае, если первый и последний символ совпадают, то сделаем сдвиг строки и сожмем эти два символа, в ином случае ничего не делаем.

Покажем, что данный алгоритм дает строку минимальной длины и что он использует минимальное количество сдвигов. Для доказательства первого вначале поймем, что получившуюся строку нельзя больше сжать. После работы алгоритма осталась строка, у которой первый и последний символ различаются. Если после нескольких сдвигов получилось сжать строку, значит сдвигов было использовано минимум 2, так как перед этим должны первый и последний символ совпадать, а после алгоритма они различаются. Значит до этого было два подряд одинаковых символа, которых алгоритм должен был сжать, противоречие.

Теперь поймем, почему другой алгоритм не мог дать строку меньшей длины. Представим строку в следующем виде : разобьем на зацикленные отрезки равных символов. Тогда любые действия со строкой не уменьшают количества таких отрезков. Алгоритм делает строку длиной, равной количеству отрезков, поэтому строка минимальной длины.

Осталось доказать, почему в строках, в которых после сжатия первый и последний символ совпадают, ответ не может быть 0. Это следует из того, что, добавив сдвиг, можно уменьшить длину строки.

```
n = int(input())
s = input()
if s[0] == s[-1]:
    if s.count(s[0]) == n:
        print(0)
    else:
        print(1)
else:
    print(0)
```

Задача G. Недостающее число

В данной задаче существует несколько способов решения, разобрано самое простое для написания.

Заметим следующее : нам заранее известна сумма чисел, которые должны быть в подарке. Можно её посчитать, используя цикл, или по формуле $\frac{n(n+1)}{2}$. Также можем посчитать сумму чисел, которые не потерялись. Тогда разница между исходной суммой и суммой набора будет в точности числом, которое было потеряно.

```
n = int(input())
sum1 = n * (n + 1) // 2
sum2 = 0
for i in range(n - 1):
    x = int(input())
    sum2 += x
print(sum1 - sum2)
```

Задача H. Температурный гардероб

Для того чтобы решить эту задачу давайте промоделируем процесс, описанный в задаче. Будем перебирать день, а также для текущего комплекта одежды будем поддерживать минимальную и максимальную температуру, в которую мы носили его. Как только разница между этими двумя температурами становится больше d , Стасу нужно переодеться.

```
n = int(input())
d = int(input())
mn = int(input())

mx = mn
ans = 0
for i in range(n - 1):
    t = int(input())
    mn = min(mn, t)
    mx = max(mx, t)
    if mx - mn > d:
        ans += 1
        mn = 1e9
        mx = -1e9
print(ans)
```