Задача 1. Беговая дорожка

К моменту второй встречи оба бегуна пробегут суммарно расстояние $2s$, поэтому время, прошедшее до их второй встречи, равно $2s/(u+v)$.

Ответ необходимо записать в виде:

2 * s / (u + v)

или в виде любого другого эквивалентного выражения.

Задача 2. Торты

Наилучшим будет следующий порядок.

5 3 6 1 4 2

При таком порядке все торты будут изготовлены за 180 минут.

Эта задача называется «задачей о двух станках». В общем случае она решается так. Упорядочим все торты по значению минимума из времени работы кондитера и упаковщика. Рассмотрим торты в таком порядке. Если минимум достигается на времени работы кондитера, то этот торт будет обработан первым из оставшихся, а если на времени работы упаковщика — то последним. В данном примере минимальное время работы кондитера или упаковщика имеют два торта: 2 $(20; 10)$ и 5 $(10; 35)$. Торт 5 должен быть изготовлен первым, торт 2 — последним. Затем идёт торт 3 $(15; 15)$. Его нужно поставить после 5 (а можно и перед 2). Затем торт 4 $(35; 20)$. У него минимально время работы упаковщика, поэтому его ставим перед последним тортом. Затем торт 6 $(35, 45)$, у него минимальное время работы кондитера, ставим его за тортом 3. Остался торт 1, который встанет в середину.

Задача 3. Много пирожных

Правильный ответ.

1 10
4 4
10 2
50 50
25 51

Для первых трёх примеров можно перебрать все варианты руками.

Для четвёртого примера рассматриваются следующие варианты: $1 \cdot 99$ (взяли по 99 пирожных одного вида), $2 \cdot 98$ (взяли по 98 пирожных двух видов), $3 \cdot 97$, ..., $99 \cdot 1$ (взяли по одному пирожному 99 видов).

Наибольшее из этих величин будет $50 \cdot 50$. Рассмотрим какой-то другой вариант, его можно представить в виде $(50 + x) \cdot (50 - x)$, где x — положительное или отрицательное число. Тогда $(50 - x) \cdot (50 + x) = 50^2 - x^2 < 50^2$, то есть наибольшим это значение будет при $x = 0$.

В последнем примере рассматриваются варианты $25 \cdot 51$, $24 \cdot 53$, $23 \cdot 51$, ..., $1 \cdot 99$. Пусть мы выбрали $25-x$ видов пирожных, тогда каждого вида было выбрано $51+2x$ пирожных. Общее число выбранных пирожных равно $(25 - x) \cdot (51 + 2x) = (25 - x) \cdot (50 + 2x) + 25 - x = 2(25^2 - x^2) + 25 - x$. Поскольку $x \geqslant 0$, то максимум этой величины будет при $x = 0$, т.е. ответ «25 51».

Задача 4. Код, исправляющий ошибки

Подобные коды широко используются при передачи информации, при кодировании данных и т.д.

Можно построить код мощности 8 длины 6, исправляющий одну ошибку. Возможный пример такого кода:

000000
001011
010101
100110
110011
101101
011110
111000

В теории кодирования доказывается, что код большей мощности с такими параметрами построить нельзя.

Задача 5. Метро

Минимальное время, в течение которого на первом пути можно было наблюдать n поездов равно $n+a(n-1)$ (n поездов стоят на платформе по одной минуте, и $n-1$ интервал между ними по a минут). Максимальное время, в течение которого на первом пути можно было наблюдать ровно n поездов, равно $n+a(n+1)$ (добавились два интервала по a минут). Таким образом, время нахождения Тани на платформе принадлежит отрезку $[n+a(n-1); n+a(n+1)]$.

Аналогично, Таня наблюдала m поездов на второй платформе, поэтому время нахождения Тани на платформе принадлежит отрезку $[m+b(m-1); m+b(m+1)]$.

Необходимо найти пересечение данных отрезков и вывести левую и правую границы пересечения. Если пересечение пусто, то нужно вывести число -1.

Пример решения.

```
a = int(input())
b = int(input())
n = int(input())
m = int(input())
l = max(n + (n - 1) * a, m + (m - 1) * b)
r = min(n + (n + 1) * a, m + (m + 1) * b)
if l <= r:
    print(l, r)
else:
    print(-1)
```

Задача 6. Площадь

Напишем перебор по ответу: циклом увеличиваем значение ширины дорожки, пока количество плиток в дорожке не будет превосходить t. Пример такого решения.

```
n = int(input())
m = int(input())
t = int(input())
c = 2 * (n + m) - 4
d = 0
while c <= t and c > 0:
    d += 1
    t -= c
    c -= 8
print(d)
```

В этом решении в переменной d хранится количество выложенных каёмок шириной в одну клетку, в переменной c — ширина каёмки. Для самой внешней каёмки $c = 2(n+m)-4$, каждая следующая каёмка содержит на 8 плиток меньше предыдущей.

Задача 7. Космические шахматы

Заметим, что за 3 хода можно переставить коня в соседнюю клетку. Напишем алгоритм, передвигающий коня по одной в соседнюю клетку, пока не будет достигнута нужная клетка. Пример такого решения.

```
x_finish = int(input())
y_finish = int(input())
x = 0
y = 0
while x < x_finish:
    print(x + 1, y + 2)
    print(x - 1, y + 1)
    print(x + 1, y)
    x += 1
while x > x_finish:
    print(x - 1, y + 2)
    print(x + 1, y + 1)
    print(x - 1, y)
    x -= 1
while y < y_finish:
    print(x + 2, y + 1)
    print(x + 1, y - 1)
    print(x, y + 1)
    y += 1
while y > y_finish:
    print(x + 2, y - 1)
    print(x + 1, y + 1)
    print(x, y - 1)
    y -= 1
```