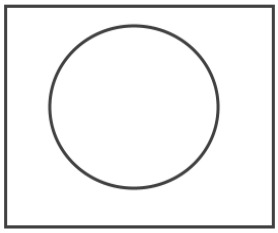
Занятие 10. Решение задач.

Учимся оформлять задачи правильно:

1. Комментарии – знак #
2. Имена переменных даем осознанные
3. Все записи оформляем пробелами

Задача 1. «Сцена». В зале квадратной формы со стороной a нужно установить сцену радиусом r. Поместиться ли сцена в зал?

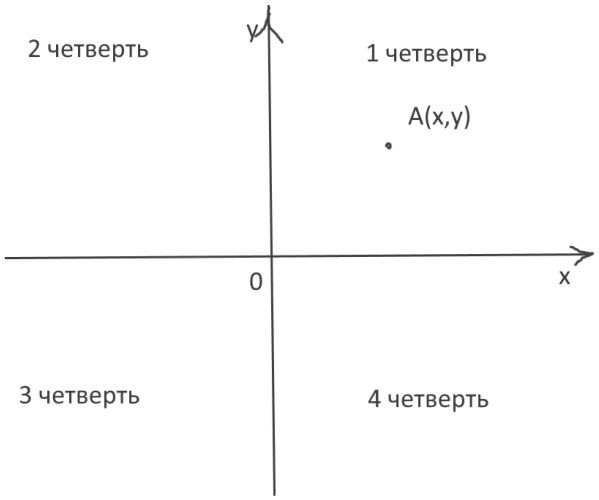


Математическая постановка – диаметр окружности должен быть меньше или равен стороне квадрата. Если равен, то окружность вписана в квадрат.

```
# Размещение окружности в квадрате
a = int(input('Введите сторону зала (квадрата)'))
r = int(input('Введите радиус сцены (окружности)'))
if 2 * r <= a:
    print('Разместить можно')
else:
    print('Разместить нельзя')
```

Задача 2. Проверить, в какой четверти находится точка с заданными координатами.

Математическая постановка:




```
x = float(input('введите координаты x'))
y = float(input('введите координаты y'))
if ((x > 0) and (y > 0)):
    print('точка находится в 1 четверти')
elif ((x < 0) and (y > 0)):
    print('точка находится во 2 четверти')
elif ((x < 0) and (y < 0)):
```

```
        print('точка находится в 3 четверти')
    elif ((x > 0) and (y < 0)):
        print('точка находится в 4 четверти')
    elif (x == 0) and (y == 0):
        print('точка находится в начале координат')
    elif y == 0:
        print('точка находится на оси y')
    else:
        print('точка находится на оси x')
```

Задача 3.

1. Найти наименьше (наибольшее) число из двух (из трех).
2. Выстроить два (три) числа по убыванию.

Два числа:

Найти наименьшее (наибольшее)

```
if a < b:
    min = a
else:
    min = b
```

Выстроить их по возрастанию – поменять их местами

```
if a > b:
    k = a
    a = b
    b = k
```

или

```
if a > b:
    a, b = b, a
```

Три числа:

Найти наименьшее (наибольшее)

```
if a < b and a < c:
    min = a
elif b < a and b < c:
    min = b
else:
    min = c
```

Выстроить их по возрастанию:

```
if a < b:
    a, b = b, a
if a < c:
    a, c = c, a
if b < c:
    b, c = c, d
```

Задача 1. Найти периметр и площадь треугольника по трем сторонам. Определить вид треугольника (равносторонний, равнобедренный, прямоугольный).

Математическая постановка.

Даны три числа. Всегда ли они определяют треугольник? Есть теорема геометрии о существовании треугольника. Например, числа 10, 3, 6 не могут определить треугольник (нельзя построить треугольник на основании трех отрезков), а числа 3, 4, 5 – дают треугольник. Теорема гласит так: каждая сторона треугольника должна быть меньше суммы двух других.

$$a < b + c,\ b < a + c,\ c < b + a$$

Это называется неравенство треугольника, оно определяет условие существования треугольника.

Если треугольник существует, то можно найти его периметр и площадь по формулам: P = a + b + c, $S = \sqrt{p * (p - a) * (p - b) * (p - c)}$, где p – полупериметр.

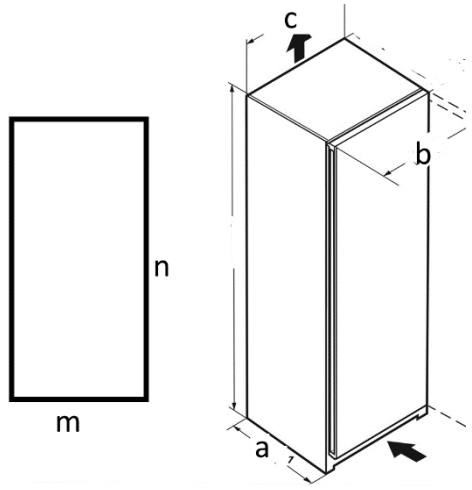
Треугольник является равносторонним, если a = b = c.
Треугольник является равнобедренным, если a = b или b = c или a = c.
Треугольник является прямоугольным, если длины его сторон удовлетворяют теореме Пифагора: $c^2 = a^2 + b^2$, но в теореме говорится, что c – гипотенуза, то есть большая сторона, а мы не определяли какая сторона большая, поэтому можно просто проверить все стороны:$a^2 = b^2 + c^2$ или $b^2 = a^2 + c^2$.
Переходим к алгоритмизации и программированию
Чтобы воспользоваться модулем, который найдет корень из числа необходимо подключить библиотеку math и модуль sqrt.

```
import math                          # подключаем библиотеку
a = int(input("Введите 1 сторону треугольник"))      # вводим три числа
b = int(input("Введите 2 сторону треугольник"))
c = int(input("Введите 3 сторону треугольник"))
if (a < b + c) and (b < a + c) and (c < a + b):      # проверяем эти числа
                                                     составят треугольник?
    print('Треугольник существует')
    P = a + b + c                                    #периметр
    pp = P /2                                        #полупериметр
    s = math.sqrt(pp * (pp - a) * (pp - b) * (pp - c))   #находим площадь
                                                      используя функцию sqrt
    print("Периметр треугольника равен ", P, "см, площадь = ", pp, "кв.см")
    if a == b and b == c:                            #проверка равенства всех сторон
```

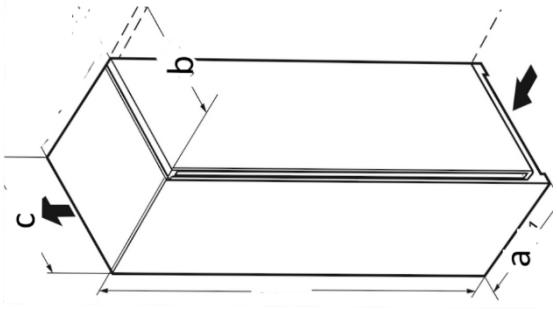
```
        print('Равносторонний')
    if a == b or a == c or b == c:                  #проверка равенства двух сторон
        print("Равнобедренный")
    if (a ** 2 == b ** 2 + c ** 2) or (b ** 2 == a ** 2 + c ** 2) or (c ** 2 == a ** 2 +
b **2):                                             #проверка теоремы Пифагора
        print('Прямоугольный')
else:                                               #неравенство треугольника не
                                                     выполняется
    print("Треугольника нет")
```

В этой задаче реализованы вложенные алгоритмы ветвления.

Задача 2. «Холодильник». Необходимо пронести холодильник в дверь. Пройдет ли холодильник? Какой стороной?
Дверной проем имеет размер m * n, холодильник – a, b, c.



<u>Математическая постановка:</u>
Холодильник можно переворачивать. То есть он пройдет в дверь стороной a*b, если (a <=n и b <=m) или (a <=m и b <=n)
Попробуем другой стороной:
(a <=n и c <=m) или (a <=m и c <=n)
Третьей стороной:
(c <=n и b <=m) или (c <=m и b <=n)

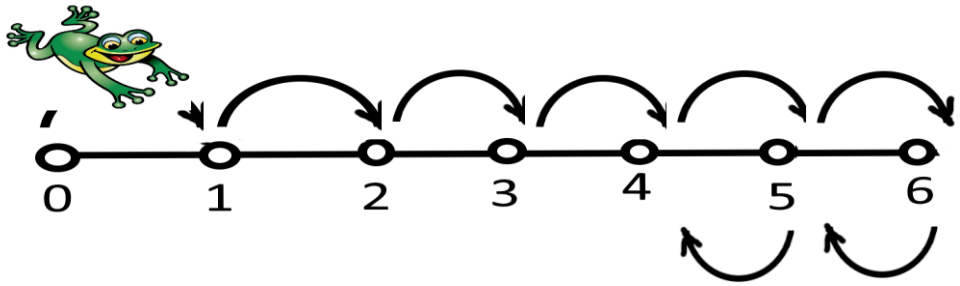


Программа имеет вид:

```
m = int(input('Введите ширину двери'))
n = int(input('Введите высоту двери'))
a = int(input('Введите ширину холодильника'))
b = int(input('Введите высоту холодильника'))
c = int(input('Введите глубину холодильника'))
if (a <= m and b <= n) or (b <= m and a <=n):
    print('Холодильник занесем сторонами ', a, b)
elif (a <= m and c <= n) or (c <= m and a <=n):
    print('Холодильник занесем сторонами ', a, c)
elif (b <= m and c <= n) or (c <= m and b <=n):
    print('Холодильник занесем сторонами ', b, c)
else:
    print('Холодильник не проходит')
```

Задача 3. (логическая). «Лягушка прыгает по камешкам». Лягушка подошла к дорожке и заметила, что на ней лежат камешки по прямой линии на одинаковом расстоянии друг от друга (всего m камешков). Лягушка начала

прыгать, сначала на первый, на второй и т.д. вправо. Допрыгнув так до последнего камешка, повернула назад, и начала прыгать влево. Допрыгнув до первого - повернула вправо и т.д. На каком камешке окажется лягушка, попрыгав n раз?



Математическая постановка. Для простоты понимания, пример m = 6, а затем обобщим наши рассуждения.
Прыгав по камешкам, лягушка может пройти полную дорожку вправо и назад, и тогда оставшиеся прыжки (неполная дорожка) будет делать вправо.
Если лягушка прыгала всю дорожку вправо, влево и опять вправо, то оставшиеся прыжки она будет прыгать влево. Это значит, что нужно проверять на четность и нечетность пройденных целых дорожек:
n // 6 % 2 == 0.
Если лягушка пропрыгает всю дорожку ту и обратно, и оставшиеся прыжки бедет осуществлять вправо, то не сложно догадаться, что она остановится на камешке с номером n % 6. А если она будет последние прыжки прыгать влево, то номер камешка 6 - n % 6.

```
m = int(input('Какая длина дорожки'))
n = int(input('Сколько раз прыгнула лягушка'))
if (n // m ) % 2 == 0:
    print('вправо', n % m)
else:
    print('влево ', m - n % m)
```

Домашнее задание:

1. «Робинзон». На необитаемом острове квадратной формы, со стороной a, живет n робинзонов. Хватает ли места всем робинзонам, если на одного робинзона необходимо k квадратных метров.