# Формальное приветствие

На занятии мы написали функцию greet():

**def** greet():   
    name = **input**()   
    **print**("Hello,␣", name)

Исправьте эту функцию так, чтобы она спрашивала у пользователя не только имя, а имя и фамилию (каждое на отдельной строке), а затем выводила официальное приветствие в форме “Здравствуйте, {**имя**} {**фамилия**}.”.

В решении вы должны только написать эту функцию, но не запускать её. Чтобы тестирующая система могла проверить её, функция должна быть названа в точности greet и не должна иметь аргументов.

## Формат ввода

greet()

*C клавиатуры вводится:*

Иван  
Смирнов

## Формат вывода

Здравствуйте, Иван Смирнов.

# Пробелы

Напишите функцию **space\_game(text)**, которая симулирует следующую игру: если количество пробелов в строке в переменной **text** является четным, то печатается: «Вы выиграли», в противном случае выводится сообщение с текстом: «Вы проиграли».

### Пример 1

| Ввод | Вывод |
| --- | --- |
| space\_game('я иду с мечем судия') | Вы выиграли |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| space\_game('Привет') | Вы выиграли |

# 

# Какая четверть?

Представьте себе координатную плоскость, образованную осями OX и OY. Оси делят плоскость на 4 равных части, которые для удобства пронумеровали следующим образом: четверть справа вверху обозначается I, и далее против часовой стрелки соответственно II, III и IV.

Напишите функцию **quarter(xcoord, ycoord)**, которая принимает в качестве параметра координаты точки на плоскости и печатает, в какой четверти она лежит (см. пример)

### Пример 1

| Ввод | Вывод |
| --- | --- |
| quarter(3, 4) | I четверть |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| quarter(-3.5, 8) | II четверть |

# 

# Треугольник?

Чтобы из трех отрезков можно было составить треугольник, необходимо и достаточно, чтобы сумма длин любых двух отрезков была строго больше третьего.

Напишите функцию **triangle(a, b, c)**, которая принимает на вход три длины отрезков и определяет, можно ли из этих отрезков составить треугольник. Ваша функция должна печатать «Это треугольник», если составить треугольник можно, и «Это не треугольник», если нельзя.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| triangle(1, 1, 2) | Это не треугольник |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| triangle(7, 6, 10) | Это треугольник |

### Пример 3

| Ввод | Вывод |
| --- | --- |
| triangle(20, 13, 17) | Это треугольник |

# Среднее значение

Напишите функцию **print\_average(arr)**, которая принимает список **целых** чисел в качестве аргумента и печатает среднее значение элементов этого списка.

В случае, когда список пустой, функция должна напечатать ноль.

Как и в прочих заданиях этого урока, в вашем решении функция должна быть определена, но не должна вызываться.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print\_average([]) | 0 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print\_average([79]) | 79.0 |

# 