**Функции**

# Формальное приветствие

Kлассная работа

макс. 1 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

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.

## Примечания

Эта и последующие похожие задачи проверяются следующим образом:  
1. В тестирующую систему вы должны отправить файл с **только**необходимой функцией. Если в файле есть код вызова данной функции, то закомментируйте его.  
2. При проверке вашей функции тестирующая система запускает программу на Python, которая осуществляет вызов вашей функции (с передачей в неё параметров).  
3. Вам необходимо точно соблюсти формат сигнатуры функции, который дается в условии.

# Привет, как тебя там?

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите функцию who\_are\_you\_and\_hello(), которая читает имя пользователя из стандартного ввода, пока он не введет его в правильном формате, а затем здоровается. Приветствие должно быть в форме: “Привет, **{имя}**!”. После вывода надо не забыть перевести курсор на новую строку.

Если пользователь ввёл имя в неправильной форме, функция должна считать его снова. Корректным считается имя, состоящее из одного слова, в котором нет символов кроме букв, первая буква заглавная, остальные — строчные.

После вывода приветствия функция должна завершить свою работу.

Обратите внимание: в вашей программе должна быть функция who\_are\_you\_and\_hello, но она не должна вызываться. Следите за тем, чтобы имя функции было написано верно.

## Формат ввода

who\_are\_you\_and\_hello()

*C клавиатуры вводится:*

Зачем тебе это знать?  
Хорошо, записывай  
Василий Пупкин  
Василий 1  
Вася1  
Вася!  
ВАСЯ  
Вася  
И тебе привет  
Михаил?  
Михаил

## Формат вывода

Привет, Вася!

## Примечания

Данную задачу вы можете решить методами, описанными в уроке по методам списков и строк. Но, к сожалению, в него нельзя было включить все возможные строковые методы, поэтому для каждого программиста крайне важным является работа с документацией. Изучите официальную документацию: [https://docs.python.org/3/library/stdtypes.html#string-\_methods](https://docs.python.org/3/library/stdtypes.html#string-methods) и найдите метод, позволяющий более эффективно решить поставленную задачу.

# Какая четверть?

Kлассная работа

макс. 2 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

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 |

## Примечания

Гарантировано, что точка не будет лежать на какой-либо из осей OX или OY.

[Открыть редактор](https://lyceum.yandex.ru/courses/468/groups/3752/lessons/2492/tasks/17583/solutions)

# Золотое сечение

Kлассная работа

макс. 2 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Вам необходимо написать функцию **golden\_ratio(i)**, которая получает число i как аргумент и печатает i-ое последовательное приближение золотого сечения.

i-ое приближения золотого сечения вычисляется как отношение (i+1)-го числа Фибоначчи к i-му.

Числами Фибоначчи называется последовательность, первые два элемента которой (нумерация с единицы) равны 1, а каждое из последующих чисел равно сумме двух предыдущих. Первые члены последовательности Фибоначчи таковы:

https://contest.yandex.ru/testsys/tex/render/MSwxLDIsMyw1LDgsMTMsMjEsMzQsNTUsXGRvdHM=.png

Таким образом первое приближение золотого сечения — 1/1, второе — 2/1, третье — 3/2, четвертое — 5/3 и т. д.

Следите за правильностью названия функции.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| golden\_ratio(1) | 1.0 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| golden\_ratio(2) | 2.0 |

### Пример 3

| Ввод | Вывод |
| --- | --- |
| golden\_ratio(4) | 1.6666666666666667 |

## Примечания

Ваша функция ничего не должна читать из стандартного ввода.  
Тестирующая система вызывает специальную программу, которая прочитает входные данные и вызовет вашу функцию.

# Скажи «пароль» и проходи

Kлассная работа

макс. 2 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите функцию ask\_password(), которая запрашивает у пользователя пароль и сверяет его со строкой, в которой записано слово “password”. Пользователю дается три попытки. Как только пароль совпал с правильным значением, функция должна выводить «Пароль принят» и игнорировать дальнейший ввод. Если с трех попыток пользователь не смог угадать пароль, функция должна вывести на экран «В доступе отказано» и игнорировать ввод новых паролей.

Обратите внимание: в вашей программе должна быть функция ask\_password, но она не должна вызываться. Следите за тем, чтобы имя функции было написано верно.

Тестирующая программа выглядит следующим образом:

ask\_password()

### Пример 1

| Ввод | Вывод |
| --- | --- |
| qwerty  1234  йцукен | В доступе отказано |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| qwerty  1234  password | Пароль принят |

# Статистики

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите функцию **print\_statistics(arr)**, которая принимает список целых чисел и печатает несколько так называемых статистик этого списка: число элементов, среднее значение, минимальный и максимальный элемент в списке, медиану списка.

Медианой называется число, больше которого ровно половина списка. В отсортированном списке из нечетного числа элементов он находится ровно посередине. Если элементов в списке четное число, то медианой является любое из значений, лежащих между двумя центральными элементами. Обычно в таких случаях выбирают среднее значение от двух центральных элементов. Мы просим вывести его.

Для пустого списка вместо всех статистик выведите нули.

При вычислении среднего значения, воспользуйтесь функцией, написанной вами ранее специально для этой цели. Одну собственную функцию можно вызывать из другой. Также рекомендуем написать отдельную функцию для вычисления медианы.

Например, вызов print\_statistics([3,5,8,4]) должен последовательно распечатать на экране все статистики:

4  
5.0  
3  
8  
4.5

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

### Пример 1

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

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print\_statistics([22]) | 1  22.0  22.0  22.0  22.0 |

## Примечания

Функция ничего не читает из стандартного ввода. Список чисел она принимает в качестве параметра, который ему передает проверяющая программа.

Напоминаем: чтобы получить отсортированный список, вы можете воспользоваться методом sort().

# Правильная скобочная последовательность

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Правильной скобочной последовательностью называется строка, состоящая только из символов «скобки» (открывающих "(" и закрывающих ")"), где каждой закрывающей скобке найдётся соответствующая открывающая. Например, () и (()()) — правильные последовательности, а (()(() или )( — нет.

Напишите функцию **bracket\_check(test\_string)**, которая проверяет, является ли поступившая на вход строка правильной скобочной последовательностью. Если да, она должна печатать YES, в противном случае — NO. Обратите внимание, что пустая строка также является правильной скобочной последовательностью.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| bracket\_check("()") | YES |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| bracket\_check("(()((") | NO |