# Функции. Передача параметров

# Электронный попугай

Kлассная работа

макс. 1 балл.

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

Боба очень любит попугаев, но у него аллергия на птиц. Поэтому он захотел создать себе электронного попугая.

Напишите функцию **parrot(phrase)**, которая будет симулировать речь попугая. Если в переменной phrase находится предложение, которое попугай слышал хотя бы один раз, то функция печатает эту фразу.

## Формат ввода

Строка

### Пример 1

| Ввод | Вывод |
| --- | --- |
| parrot("Привет!")  parrot("Привет!")  parrot("Как дела?") | Привет! |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| parrot("Привет")  parrot("Как тебя зовут?")  parrot("Привет") | Привет |

# Счёт за обед

Kлассная работа

макс. 1 балл.

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

Иван Иванович работает в крупной компании. Обычно в обеденный перерыв он ходит в местную столовую. Начальство хочет возместить расходы на питание.

Для большего комфорта сотрудников возмещение выдаётся за любые несколько дней прошедшего месяца. Напишите функцию **count\_food(days)**, которая принимает на вход массив чисел от 1 до 31 и возвращает, сколько в сумме было потрачено на обеды в эти дни. Сумма, потраченная в каждый из дней, находится в глобальном массиве **daily\_food**. Например, **daily\_food[0]** – сумма, потраченная в первый день.

## Формат ввода

Список целых чисел, каждое из которых – номера дней в месяце

## Формат вывода

Целое число – сумма, потраченная за указанные дни на питание.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| daily\_food = [0, 150, 150]  print(count\_food([1])) | 0 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| daily\_food = [0, 150, 150]  print(count\_food([2, 3])) | 300 |

# Числа в строке

Kлассная работа

макс. 1 балл.

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

Напишите функцию from\_string\_to\_list(string, container), которая принимает два аргумента: строку string, состоящую из целых чисел, написанных через пробел, и список container. Функция должна извлечь из строки числа и добавить их в конец списка.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| a = [1, 2, 3]  from\_string\_to\_list("1 3 99 52", a)  print(\*a) | 1 2 3 1 3 99 52 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| a = [77, 'abc']  from\_string\_to\_list("", a)  print(\*a) | 77 abc |

## Примечания

В тестовую систему вы должны отправить файл только с функцией from\_string\_to\_list(string, container). Если в файле есть код вызова данной функции, то закомментируйте его. Выводить на экран ничего не нужно.

[Открыть редактор](https://lyceum.yandex.ru/courses/468/groups/3752/lessons/2495/tasks/17539/solutions)

## История решений

Начало формы

# Обмен личностями

Kлассная работа

макс. 2 балл.

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

Для хитроумного приспособления, которое обменивает людей личностями, вы должны написать функцию, которая принимает список мыслей одного человека и другого, а затем обменивает их содержимым.

Функция должна называться swap(first, second). Первый список должен получить содержимое второго списка, и наоборот.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| first = [1, 2, 3]  second = [4, 5, 6]  first\_content = first[:]  second\_content = second[:]  swap(first, second)  print(first, second\_content, first == second\_content)  print(second, first\_content, second == first\_content) | [4, 5, 6] [4, 5, 6] True  [1, 2, 3] [1, 2, 3] True |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| first = [1, 2, 3]  second = [4, 5, 6, 7]  first\_content = first[:]  second\_content = second[:]  swap(first, second)  print(first, second\_content, first == second\_content)  print(second, first\_content, second == first\_content) | [4, 5, 6, 7] [4, 5, 6, 7] True  [1, 2, 3] [1, 2, 3] True |

## Примечания

В тестовую систему вы должны отправить файл только с функцией swap(first, second). Если в файле есть код вызова данной функции, то закомментируйте его. Выводить на экран ничего не нужно.

# Фрактальный список – 2

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 defractalize(fractal), которая принимает в качестве аргумента фрактальный список и удаляет из него все самоподобные части фрактала (т.е. части, эквивалентные самому фракталу).  
Фрактальный список в этой задаче может содержать в качестве элементов любые числа и сам себя (в любом числе копий).

### Пример 1

| Ввод | Вывод |
| --- | --- |
| fractal = [2, 5]  fractal.append(fractal)  fractal.append(3)  defractalize(fractal)  print(fractal) | [2, 5, 3] |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| fractal = [2, 5]  fractal.append(fractal)  fractal.append(3)  fractal.append(fractal)  fractal.append(9)  defractalize(fractal)  print(fractal) | [2, 5, 3, 9] |

## Примечания

В тестовую систему вы должны отправить файл только с функцией defractalize(fractal). Если в файле есть код вызова данной функции, то закомментируйте его. Выводить на экран ничего не нужно.

# Транспонирование

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 transpose(matrix), которая принимает матрицу N×MN×M (т.е. список из NN списков-строк по MM элементов каждый) и транспонирует её, т.е. превращает строки в столбцы и наоборот.

ii-ая строка при этом превращается в ii-й столбец.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| matrix = [[1]]  transpose(matrix)  for line in matrix:  print(\*line) | 1 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| matrix = [[1, 2], [3, 4]]  transpose(matrix)  for line in matrix:  print(\*line) | 1 3  2 4 |

## Примечания

В тестовую систему вы должны отправить файл только с функцией transpose(matrix). Если в файле есть код вызова данной функции, то закомментируйте его. Выводить на экран ничего не нужно.

# Печать фрактала

Kлассная работа

макс. 3 балл.

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

Когда встроенная функция print распечатывает список, содержащий сам себя, то вместо содержимого внутреннего списка она подставляет многоточие.

Это позволяет программе не зациклиться, пытаясь напечатать список, который за счет фрактальной структуры имеет бесконечную глубину.

Например, текстовое представление списка, состоящего из нуля, самого себя и двойки будет выглядеть так:

[0, [...], 2]

Напишите функцию fractal\_print(obj), которая также выводит фрактал на экран, но печатает не один, а два уровня вложенности:

[0, [0, [...], 2], 2]

Мы рассматриваем только простые фрактальные списки, состоящие из обычных несоставных объектов (числа, строки, логические величины) или самих себя.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| fractal = [3]  fractal.append(fractal)  fractal\_print(fractal) | [3, [3, [...]]] |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| fractal = [3]  fractal.append(fractal)  fractal.append(2)  fractal\_print(fractal) | [3, [3, [...], 2], 2] |

## Примечания

В тестовую систему вы должны отправить файл только с функцией fractal\_print(obj). Если в файле есть код вызова данной функции, то закомментируйте его. Выводить на экран ничего не нужно.