**Функции. Возращение значений из функций**

# Мелочь оставь себе

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 **take\_large\_banknotes(banknotes)**, которая будет принимать набор купюр и монет (в форме списка чисел, представляющих их номиналы), и возвращать список, в котором оставлены только крупные купюры больше (строго больше) 10 рублей.

Порядок отобранных купюр сохраните таким, какой был в исходном списке.

Вы только перечисляете купюры, которые хотите забрать, но не удаляете их из списка. Исходный список **должен**остаться неизменным.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print(\*take\_large\_banknotes([])) |  |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print(\*take\_large\_banknotes([1, 5, 500, 0.5, 2, 0.1, 10, 100, 100, 1000, 50])) | 500 100 100 1000 50 |

## Примечания

Будьте внимательны: ваши функция должна называться в точности так, как указано в задании и принимать тот же аргумент.

Вы можете вызывать написанную функцию, пока тестируете программу, но при сдаче задачи в проверяющую систему вызывать её не нужно.

# Месяц/Month

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, которая принимает номер месяца и язык (русский или английский), а возвращает его название.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print(month\_name(3, "en")) | March |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print(month\_name(3, "ru")) | март |

# Среднее значение – 2

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

В уроке по теме «Функции» вы писали функцию для вычисления среднего арифметического списка чисел.

Сделайте аналогичную функцию average(values), только на этот раз результат она должна возвращать, а не печатать на экране.  
Как и прежде, если список пустой, функция должна возвращать ноль.  
Возврат значения для случаев пустого и непустого списков сделайте отдельными операторами return.

## Формат ввода

Вызов

average([1, 5, 500, 10])

## Формат вывода

должен возвращать

129

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print(average([1, 2, 3, 4, 5])) | 3 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print(average([-5, 2])) | -1.5 |

# Секретные материалы

Kлассная работа

макс. 2 балл.

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

Ваша программа работает с архивом документов. Каждый документ состоит из списка строк, представляющих собой страницы текста.

Ваша задача — написать функцию print\_document(pages), которая будет просто распечатывать переданный документ страница за страницей. Иногда в документе попадаются секретные страницы. Они всегда идут в конце, а первая страница секретной части начинается словом «Секретно». Секретную часть печатать не нужно, надо остановиться на последней незасекреченной странице.

В случае, когда секретных страниц не было, выведите в конце: «Напечатано без купюр». Если секретные страницы были, выведите «Дальнейшие материалы засекречены».

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print\_document(["Обычная страница", "И еще страница", "Секретно Вот этот вот текст не показывать", "Никому", "Никогда"]) | Обычная страница  И еще страница  Дальнейшие материалы засекречены |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print\_document(["Пустой трёп", "который", "никому не интересен"]) | Пустой трёп  который  никому не интересен  Напечатано без купюр |

# Ход конём

Kлассная работа

макс. 3 балл.

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

Вам дана строка, содержащая координату клетки на шахматном поле (например, “A2”).

Напишите функцию possible\_turns(cell), которая будет по такой строке возвращать *список*координат (в такой же текстовой форме) всех клеток, на которые может попасть конь с указанной клетки за один ход. Не забудьте, что за пределы шахматного поля коню ходить нельзя.

Рекомендация:

Cделайте две вспомогательные функции, которые будут по строке делать целочисленные координаты и наоборот. Например, клетке “A2” можно сопоставить кортеж (1,2), а кортежу (8,8) — клетку “H8”. То есть, одна функция принимает строку, а возвращает кортеж из двух чисел. Другая функция принимает кортеж, возвращает строку.

Также полезно сделать функцию, которая по кортежу целочисленных координат сообщает, находится ли это клетка внутри шахматного поля или снаружи.

Проверяется только функция possible\_turns. Вспомогательные функции не проверяются, но они помогут вам написать простое и компактное решение.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print(possible\_turns("B1")) | ['A3', 'C3', 'D2'] |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print(possible\_turns("H8")) | ['F7', 'G6'] |

## Примечания

Возвращаемый список нужно отсортировать по алфавиту.

# Опоздание

Kлассная работа

макс. 3 балл.

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

Представьте себе такую неприятную ситуацию: вы просыпаетесь утром и обнаруживаете, что проспали и можете опоздать на занятия в лицее. Вы настолько не хотите опоздать, что готовы пожертвовать завтраком и выйти из дома прямо сейчас. До остановки автобуса вам бежать 5 минут, автобус будет ехать до лицея еще 15 минут (вам так повезло, что автобус ходит от вашей остановки до самых дверей лицея). Кроме того, у вас есть приложение Яндекс.Транспорт, которое показывает, через сколько минут на остановку придет нужный автобус. Теперь вам нужно вычислить, через сколько минут нужно выйти из дома, чтобы успеть на занятия в лицее.

На вход вашей функции **late(now, classes, bus)** подаются три аргумента: текущее время (в формате hh:mm, например, 9:15), время начала уроков (в том же формате hh:mm) и список из чисел, в котором находятся данные о времени прибытия автобуса (т. е. кол-во минут, через которое автобус подойдет к остановке; как вы наверняка знаете, время в приложении отсортировано по возрастанию, например “1 4 15”). Помните, с текущего времени вам нужно 5 минут, чтобы добежать до остановки, и еще 15 минут – чтобы доехать до лицея. Если вы никак не успеваете вовремя на занятия, функция должна вернуть строку “Опоздание”. В противном случае, найдите автобус, который отходит от остановки позже всех (вдруг вы даже успеете позавтракать :), функция должна вернуть, через сколько минут вам надо выйти из дома (в формате "Выйти через n минут").

Например: если сейчас 9:00, а занятия начинаются в 10:00, и автобусы приходят через 5, 15 и 25 минут, то вас устроит самый последний из них, и выйти из дома вам нужно через 25-5=20 минут.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| print(late('12:00', '12:40', [0, 1, 4, 6, 25])) | Выйти через 20 минут |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| print(late('9:20', '9:35', [4, 25, 30])) | Опоздание |

## Примечания

Данная задача дополнительно проверяется преподавателем.

# Поиски возвышенного

Kлассная работа

макс. 3 балл.

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

В этой задаче вы будете работать с картой высот для прямоугольного кусочка местности.

Местность разбита на небольшие квадраты, и для каждого квадрата записана высота его центра над уровнем моря. Эти высоты записаны в матрицу (прямоугольную таблицу).

Напишите функцию, которая найдет координату (номер ряда и номер столбца) самого высокого участка и вернет их из функции так, чтобы можно было использовать множественное присваивание:  
row, column = find\_mountain(heightsMap)

Если есть несколько участков карты, имеющих максимальную высоту, то допустимо вернуть любой из них.

Пример: вызов find\_mountain([[1,3,1],[3,2,5],[2,2,2]]) вернёт кортеж (1,2), соответствующий горе высоты 5.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| a = [[1, 3, 1], [3, 2, 5], [2, 2, 2]]  print(find\_mountain(a)) | (1, 2) |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| a = [[2, 4, 2, 3, 2], [3, 2, 3, 4, 3]]  row, col = find\_mountain(a)  print(a[row][col]) | 4 |