**Лабораторная работа 1**

## Примечания

Не забудьте добавить в начало скрипта:

#! /usr/bin/env python

Сделайте ваши файлы исполняемыми, набрав в консоли:

chmod +x <filename>

Для этой лабораторной работы не помещайте суффикс .py в имена программ; Для всех последующих добавляйте.

## Программы для написания

* Скрипт называющийся hello который выводит "hello, world!".
* Скрипт называющийся solve\_quadratic\_equation который вычисляет решения для произвольных квадратичных уравнений. Напомним, что уравнения квадратичной формы a\*x^2 + b\*x + c = 0, где x — переменная, которую вам нужно найти и b и c — действительные числа, которые вы вводите. Вашей программе необходимо подключить библиотеку math. Скрипт будет вызван следующим образом:
* solve\_quadratic\_equation 2 3 4

(где a = 2, b = 3, и c = 4) и выведет:

Quadratic equation: 2\*x^2 + 3\*x + 4 = 0

Solution 1: <whatever>

Solution 2: <whatever>

Если решения являются комплексными, следует печатать, что нет никаких реальных решений. Выведите решение в числах, а не в формулах. Предположим, что все аргументы являются числами и что a всегда не будет равно нулю.

Используйте float для преобразования строки, соответствующие аргументы командной строки для a, b и c для чисел с плавающей запятой.

**ПРИМЕЧАНИЕ**: Убедитесь, что, если пользователь вводит неверные аргументы (например, слишком мало или слишком много аргументов или аргументы, не являющиеся числами), скрипт завершит работу с сообщением об ошибке. Проверку на не числа лучше всего делать с обработкой исключений.   
Подсказка: вызов float() или string.atof() на аргумент, который не может быть преобразован в тип float вызывает исключение ValueError.

**[Расширенный]**: Не думайте, что все аргументы являются числами; Выведите сообщение об ошибке, если любой из них не являются числами.

**[Расширенный]**: Сделайте для комплексных решений. Обратите внимание, что python обрабатывает комплексные числа, но не функция sqrt. Модуль «cmath» может оказаться полезными.

**[Расширенный]**: Сделайте также для случаев, где а = 0, b = 0, и/или c = 0.

* Скрипт под названием count\_lines который читает файл (filename аргумент командной строки) и выводит имя файла и сколько строк там находятся. Не используйте readlines метод для файлов, потому что файл считывается целиком в память, что является плохой идеей для больших файлов. Опять же убедитесь, что если скрипт вызывается, например, с слишком большим или маленьким количеством аргументов, то выведите сообщение перед завершением работы скрипта.   
  **[Расширенный]**: Заставить его работать для произвольного количества имен файлов в командной строке. В этом случае выведите отдельной строкой имя файла и количество строк в этом файле.
* Скрипт называющийся capitalize который читает все строки в фале, меняет их регистр на прописной, и записывает их в другой файл. Командная строка должна выглядеть так:

capitalize myfile myfile\_caps

где myfile название читаемого файла и myfile\_caps название нового файла в который вы записываете.

Используйте upper функцию string модуля для смены регистра. Обратите внимание на количество аргументов при вызове скрипта. Так же убедитесь что вы закрываете все файлы, которые были открыты.