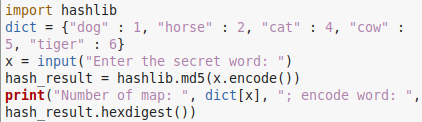
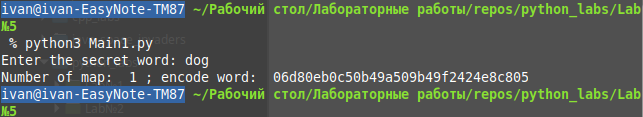
Лабораторная работа №5

“Профилиорвание и оптимизация кода”

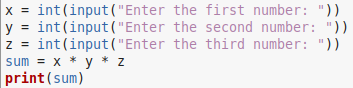
Задание 1.

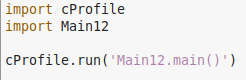
1. Примера с выводом хеша по алгоритму MD5.

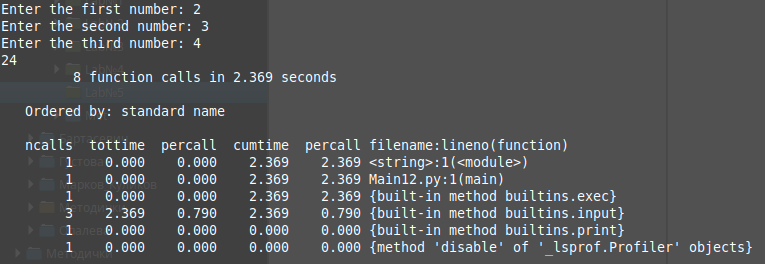




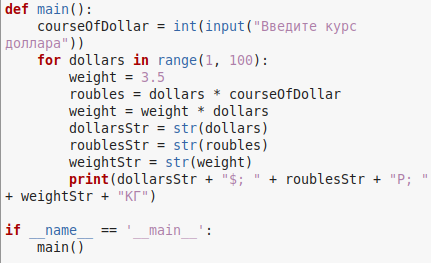
1. Примера с использованием регулярного выражения.

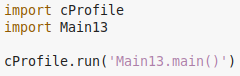


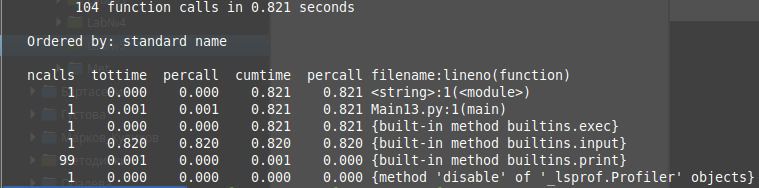




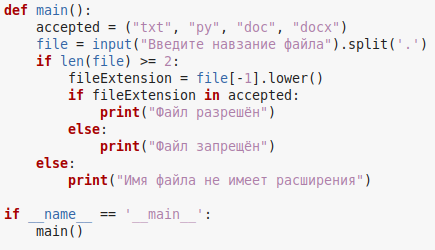
1. Первой циклической задачи из ЛР 1-4.

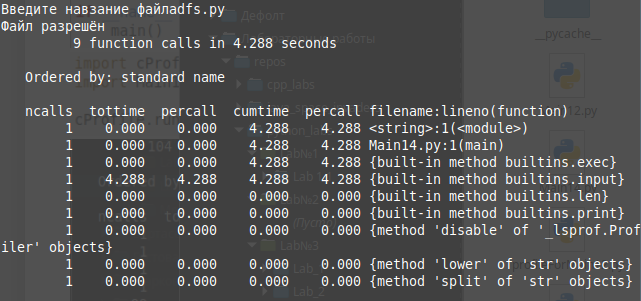




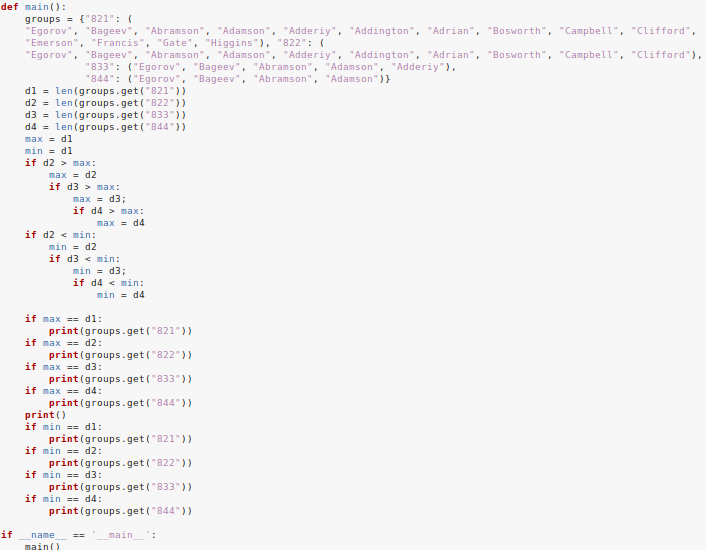


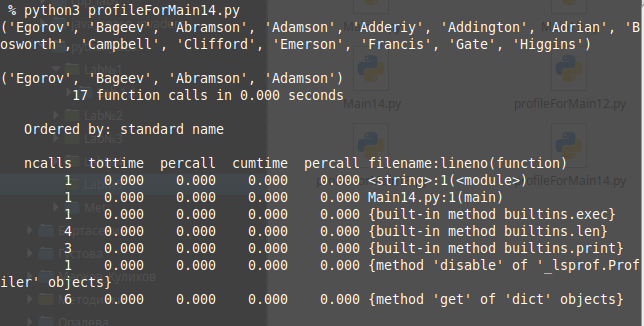
1. Последней задачи из раздела «Строки» ЛР 5-7.





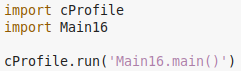
1. Последней задачи из раздела «Списки» ЛР 5-7.

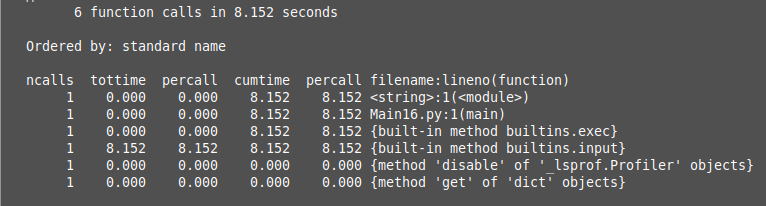




1. Последней задачи из раздела «Словари» ЛР 5-7.

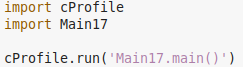


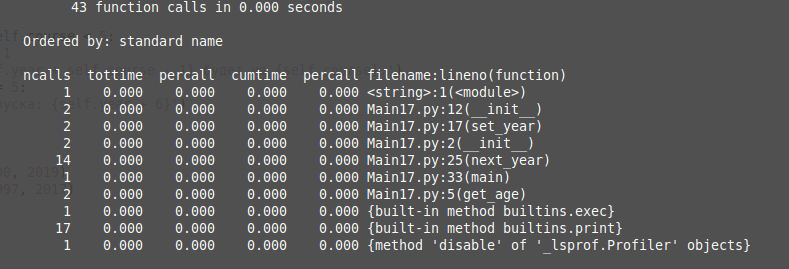




1. Второй задачи на классы с наследованием из ЛР 8-10.



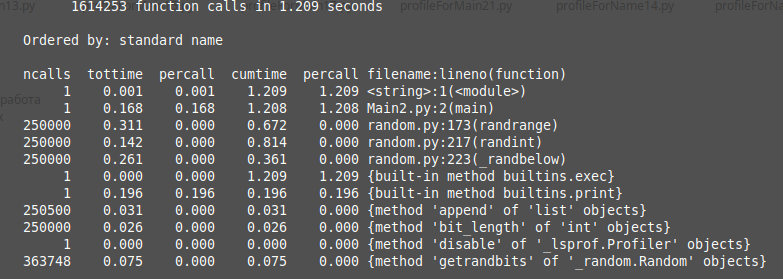




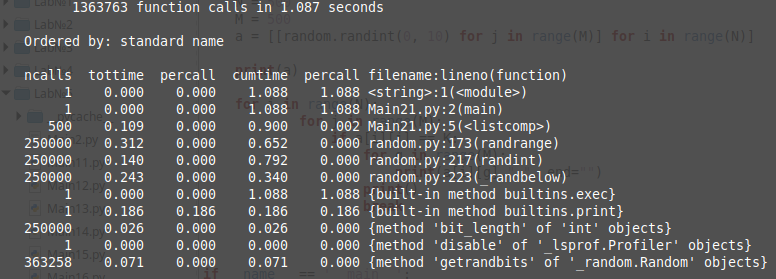
Задание 2.

В задаче на двумерный список вместо генератора использовать цикл с поэлементным добавлением элементов в список. Выполнить профилирование кода. Вычислить разницу производительности в процентах (при необходимости увеличить размерность списка).

Без генератора массива



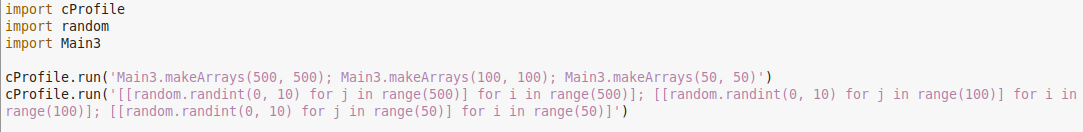
С генератором массива

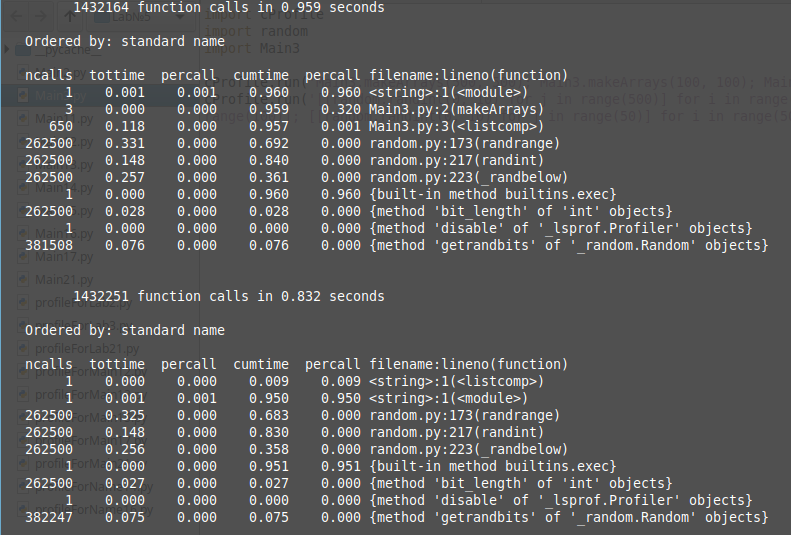


Одна и та же задача решилась с генератором массива на 0.122 секунды быстрее. В процентном соотноешнии, благодаря профилированнию, мы смогли оптимизировать код генератором на 10%.

Задание 3

Изучить документ «[Дополнение к ЛР14-17.docx](Дополнение%20к%20ЛР14-17.docx)». Решение задачи с генератором двумерного списка оформить в виде функции, возвращающей список, со входными параметрами в виде числа строк и столбцов. Вызвать функцию несколько раз в основной программе. Оформить код согласно правилам стиля. Выполнить профилирование кода. Сравнить время на вызов функции с временем, полученным для этой задачи в п.1.





Функциями выполняется медленее, чем просто генерировать массив на 12%.