**Модуль №7. «Углубленный Python»**

**Задание №1.1**

Решение задач прислать ссылкой на github репозиторий.

Структура репозитория

Каждый набор задач должен лежать в папке, «practice\_номер\_набора\_задач». В наборе могут быть простые, сложные и задачи уровня мини-приложений. Простые задачи можно написать в одном файле (**имя файла script.py**), каждую сложную в отдельном файле (**подумайте над наименованием файлов!**). Задача считается сложной, если стоит символ **«\*»**! Задачи уровня мини-приложения помечаются «\*\*» и оформляются отдельной папкой с наборов файлов: файл скрипта, входных данных и готовый файл отчета (выходные данные).

корень: python-high-difficulty-practices

папка: practice\_1

файл: main\_2.py

Простые функции (ПОВТОРЕНИЕ-ЗАКРЕПЛЕНИЕ)

**Задача №1.** Разработать функцию на вход в которую подается массив вещественных чисел. Найти максимальный элемент и его координату. Если в массиве несколько максимальных элементов, найти только первый из них. Результатом выполнения функции будет tuple, в формате: (максимальный элемент, координата максимального элемента).

**Задача №2.** Разработать функцию на вход в которую подаются: исходный массив целых чисел, левая и правая границы инвертирования (по умолчанию левая = 0, правая = -1). Провести операцию инвертирования, при этом не изменяя исходный массив. Результатом выполнения функции будет инвертированный массив.

**Задача №3.** Разработать функцию (главная) для нахождения суммы всех элементов массива вещественных чисел по заданному условию. На вход в функцию передается массив вещественных чисел, и ссылка на служебную функцию, которая будет реализовывать условие и возвращать значение типа bool. Результатом выполнения главной функции будет число, сумма элементов массива удовлетворивших условию.

**Задача №4 \*.** Разработайте функцию (главная) на вход в которую передаются: массив вещественных чисел, ссылка на служебную функцию, которая будет реализовывать признак сортировки и возвращать значение типа bool. Главная функция должна отсортировать полученный массив по переданному признаку сортировки. Сортировку реализовать методом «Пузырька» (для тех, кто хочет, можно еще и методом простых вставок, в отдельной функции). Исходный массив переданный на вход не должен быть изменен. Результатом выполнения функции будет новый отсортированный массив.

Обработка данных (ПОВТОРЕНИЕ-ЗАКРЕПЛЕНИЕ)

**Задача №5. Анализ трат в кафе \*\***

*Описание*  
Разработайте программу, которая поможет владельцу кафе проанализировать расходы за последние 12 месяцев. Вам предоставлены данные о расходах по различным категориям. *Программа должна:*

* Вычислить общую сумму расходов и средний расход по каждой категории;
* Определить типы трендов для каждой категории;
* Спрогнозировать будущие расходы на основе текущих тенденций.

*Дополнительные функции:*

1. Функция прогнозирования расходов. Прогнозируемый расход рассчитывается как сумма всех трендов.
2. Функция вычисления трендов расходов. Тренд - в данном контексте относится к общему направлению изменения расходов кафе по каждой категории за определенный период. Тренд может быть возрастающим, убывающим или стабильным. Тренд считается стабильным, если он равен расходам за последние два месяца по отдельности. Тренд считается возрастающим, если он больше расходов за последние два месяца по отдельности. Тренд считается убывающим, если он меньше расходов за последние два месяца по отдельности. Тренд вычисляется как средний расход по каждой категории.

*Служебные функции:*

* Функция чтения данных из файла;
* Функция вычисления средних значений;
* Функция записи данных в файл.

*Тестовый пример №1*

Входные данные:

Файл: **“cafe\_expenses\_extended.csv”**

GitHub: <https://github.com/dante-pol/python-course.git>

Выходные данные:

Файл: **“cafe\_expenses\_extended\_report.txt”**

Углубленный отчет о расходах кафе за последние 12 месяцев

Общие расходы: X руб.  
Средние расходы по категории <наименование категории>;  
…  
Средние расходы по категории <наименование категории>;

Тренд расходов (возрастание/убывание) по категории <наименование категории>;   
…  
Тренд расходов (возрастание/убывание) по категории <наименование категории>;

Прогнозируемые расходы на следующий год: Y руб.

**Задача №6. Анализ посещаемости музея \*\***

*Описание*

Создайте программу для анализа посещаемости музея за последний год. Вам предоставлены данные о ежедневной посещаемости.

*Программа должна:*

* Вычислить общее количество посетителей за год и среднее количество посетителей в месяц;
* Определить пиковые и непиковые дни, а также проанализировать влияние праздников на посещаемость;
* Рассчитать процентное изменение посещаемости между месяцами;

*Дополнительные функции:*

1. Функция анализа пиковых и непиковых дней. Пиковые дни определяются как дни с количеством посетителей выше среднего. Непиковые дни - соответственно, с количеством дней меньше среднего.
2. Функция определения влияния праздников на посещаемость. Сравните среднее количество посетителей в праздничные дни со средним количеством посетителей в обычные дни. Важно также учесть предшествующие и последующие дни, так как праздники могут влиять на посещаемость не только непосредственно в день праздника, поэтому в качестве дельты берется количество за крайний левый и правый дни, и сам за сам праздничный день.
3. Функция вычисления процентного изменения посещаемости.
   1. Расчет месячной посещаемости:
      * Для каждого месяца нужно суммировать количество посетителей, чтобы узнать общую месячную посещаемость.
      * Затем, рассчитать среднее количество посетителей в месяц.
   2. Вычисление процентного изменения:
      * Процентное изменение между месяцами можно рассчитать, взяв разницу в количестве посетителей между двумя месяцами, деленную на количество посетителей в первом месяце, и умноженную на 100.

Это даст изменение в процентах от месяца к месяцу, показывая, где был рост, а где - снижение посещаемости.

**Служебные функции:**

* Функция чтения данных;
* Функция обработки дат;
* Функция вычисления средних значений;
* Функция записи данных в файл.

*Тестовый пример №1*

Входные данные:

Файл: «**museum\_attendance\_extended.csv»**

GitHub: <https://github.com/dante-pol/python-course.git>

Выходные данные:

Файл: «**museum\_attendance\_extended\_report.txt»**

Детальный отчет о посещаемости музея за последний год

Общее количество посетителей: N  
Среднее количество посетителей в месяц: AVG

Пиковые и непиковые дни: [список дат]

Влияние праздников и специальных событий: [анализ]

Процентное изменение посещаемости между месяцами: [детализация]