

Лабораторная работа №2

Тема: «Работа с вводом и выводом данных, чтение и запись файлов».

Требования к выполнению лабораторной работы №2

1. Изучите теоретическую часть ко второй лабораторной работе:
 - a. Работа с вводом и выводом данных, чтение и запись файлов (notebook_2.ipynb)
 - b. Лекция №2.
2. Создайте новый проект.
3. Запустите примеры из лабораторной работы.
4. Выполните задание согласно вашему варианту:
 - a. Вычислите свой вариант (*согласно формуле ниже*).
Если сделали не свой вариант => работа не засчитывается.
 - b. Каждое задание представляет собой отдельный скрипт формата:
lab_{номер_ЛР}_{номер_задания}_{номер_варианта}.py, пример:
lab_2_1_2.py
 - c. Отправьте выполненное задание в ОРИОКС (*раздел Домашние задания*).

Формат защиты лабораторных работ:

1. Продемонстрируйте выполненные задания.
2. Ответьте на вопросы по вашему коду.
3. При необходимости выполните дополнительное (*дополнительные*) задания от преподавателя.
4. Ответьте (*устно*) преподавателю на контрольные вопросы.

Список вопросов

1. Перечислите основные режимы работы с файлами. Чем они отличаются?
2. Перечислите основные методы считывания данных из текстового файла. Чем они отличаются?
3. Для чего нужен параметр *encoding* в функции *open*?
4. В чём отличие текстового файла от бинарного?
5. Какие данные обычно хранят в формате *csv*?
6. Специфика работы библиотеки *pickle*.
7. Отличия *pickle* от JSON.

Задания

Во всех заданиях необходимо проверять корректность вводимых данных и выводить соответствующие сообщения об ошибках.

№ Варианта = номер_студенческого % 3 + 1

Вариант №1

Задание №1. Сгенерируйте файл `ip.log` состоящий из 10 000 уникальных IPv4 адресов. Каждый IP адрес должен находиться на новой строке.

Пример:

```
192.168.1.1
95.162.15.20
...
10.0.0.1
```

Задание №2. Пользователь с клавиатуры вводит маску подсети [\[1\]](#). Напишите скрипт, который считывает файл `ip.log` (из первого задания), для каждого IP адреса применяет маску подсети и записывает в файл `ip_solve.log` адреса сети [\[2\]](#).

Пример:

IP-адрес	192.168.1.2 (формат из файла) 11000000 10101000 00000001 00000010 (binary)
Маска подсети	255.255.254.0 (ввод с клавиатуры) 11111111 11111111 11111110 00000000 (binary)
Адрес сети	192.168.0.0 (запись в файл) 11000000 10101000 00000000 00000000 (binary)

Задание №3. В файле *players.csv* записан (формируется студентом вручную) протокол турнира в формате:

```
Спортсмен;Количество побед;Доп. показатель
Иванов;10;256
Петров;30;1000
Медведев;30;1100
...
Сидоров;20;300
```

Необходимо записать в текстовый файл *results.csv* результаты турнира.

Распределение мест зависит от следующих показателей:

1. Количество побед (чем больше, тем выше участник турнира).
2. Дополнительный показатель (учитывается для участников, набравших одинаковое количество побед, чем выше показатель, тем выше участник).

Пример:

Входной файл *players.csv* (заголовок с названиями полей – обязателен)

```
Спортсмен;Количество побед;Доп. показатель
Иванов;10;256
Петров;30;1000
Медведев;30;1100
Сидоров;20;300
Уткин;10;256
Васин;5;100
```

Выходной файл *results.csv* (заголовок с названиями полей – обязателен):

```
Спортсмен;Место
Медведев;1
Петров;2
Сидоров;3
Иванов;4
Уткин;4
Васин;6
```

Примечание: если участники имеют одинаковое количество побед и доп. показателей, то они делят место.

Задание №4. Внутри запускаемого проекта (*должен использоваться относительный путь*) создайте директорию `example`. Внутри директории `example` сгенерируйте 1000 файлов. Название файлов должно содержать латинские буквы и цифры.

Задание №5. Пользователь с клавиатуры вводит строку. Необходимо вывести (*в консоль*) количество файлов в директории `example` (из задания №4), название которых содержит подстроку, введенную пользователем.

Задание №6. Создайте файл `article_rus.txt` и заполните его текстом любого художественного произведения на русском языке (*ограничение – размер файла должен быть не менее 10 Кб*). Необходимо определить частоту (*в долях*) повторяемости каждой кириллической буквы в тексте (*остальные символы игнорировать*), отсортировать в порядке убывания частоты, результат записать в файл `article_rus_solve.txt` в формате: `{буква}: {частота}`.

Пример:

о: 0.095

е: 0.074

...

Примечание: символы в нижнем и верхнем регистре учитывать как один символ.

Вариант №2

Задание №1. Сгенерируйте файл `mask.log` состоящий из всех [ВОЗМОЖНЫХ масок](#) подсети. Каждая маска должна находиться на новой строке.

Пример:

```
255.255.255.255
255.255.255.254
...
000.000.000.000
```

Примечание: маски должны генерироваться скриптом, а не заданы в коде (в любом виде).

Задание №2. Пользователь с клавиатуры вводит IPv4 адрес. Напишите скрипт, который считывает файл `mask.log` (из первого задания), для каждой строки применяет маску подсети и записывает в файл `ip_solve.log` адреса сети [\[2\]](#).

Пример:

IP-адрес	192.168.1.2 (ввод с клавиатуры) 11000000 10101000 00000001 00000010 (binary)
Маска подсети	255.255.254.0 (формат из файла) 11111111 11111111 11111110 00000000 (binary)
Адрес сети	192.168.0.0 (запись в файл) 11000000 10101000 00000000 00000000 (binary)

Задание №3. В файле `store.csv` записана (формируется студентом вручную) информация о товарах на складе:

```
Товар;Категория;Стоимость
Шоколад;Сладости;200.50
Яблоко;Фрукты;99.99
...
Молоко;Молоко;300
```

Необходимо записать в текстовый файл `categories.csv` информацию о категориях: *Категория => Общая стоимость товаров категории.*

Пример:

Входной файл `store.csv` (заголовок с названиями полей – обязателен)

```
Товар;Категория;Стоимость
Шоколад;Сладости;200.50
Яблоко;Фрукты;99.99
Бананы;Фрукты;10
Молоко;Молоко;300
```

Выходной файл `categories.csv` (заголовок с названиями полей – обязателен):

```
Категория;Стоимость
Сладости;200.50
Фрукты;109.99
Молоко;300
```

Примечание: в столбце «Стоимость» все значения > 0 и имеют не более двух знаков после запятой.

Задание №4. Внутри запускаемого проекта (должен использоваться относительный путь) создайте директорию `example`. Внутри директории `example` сгенерируйте 100 файлов с различными расширениями (на ваш выбор, не менее 10 различных расширений, размер файла не важен).

Задание №5. Пользователь с клавиатуры вводит расширение файла. Необходимо вывести (в консоль) количество файлов в директории `example` (из задания №4), расширение которых совпадает с заданным пользователем.

Задание №6. Создайте файл `article.txt` и заполните его текстом любого художественного произведения. Необходимо определить частоту повторяемости каждой буквы в тексте, отсортировать в порядке убывания частоты, результат записать в файл `article_solve.txt` в формате: {буква}: {частота}.

Пример:

о: 0.095

б: 0.074

я: 0.01

...

Примечание:

1. Ограничение на размер файла: не менее 10 Кб.
2. Определять частоту букв в долях.
3. Считать только буквы русского и английского алфавитов, остальные символы игнорировать.
4. Символы в нижнем и верхнем регистре учитывать как один символ.

Вариант №3

Задание №1. Сгенерируйте файл `ip.log` состоящий из 10 000 уникальных IPv6 адресов. Каждый IP адрес должен находиться на новой строке.

Пример:

`e6f7:25a4:f482:694b:931f:661:b024:341`

`ef30:449e:2bfb:6fe7:f21e:262a:48dd:f63e`

`7e6a:b99f:6c29:edde:8f2b:244e:1acb:88e5`

...

`cf9b:13c7:7ba:c074:ebec:b4eb:22f7:ee1b`

Задание №2. Пользователь с клавиатуры вводит IPv6 адрес. Напишите скрипт, который считывает файл `ip.log` (из первого задания), сравните (*) каждый IPv6 адрес с введенным из клавиатуры и запишите результат в файл `ip_solve.txt`.

(*) – сравнивать IPv6 адреса необходимо с учетом возможных сокращений [\[3\]](#).

Например:

Адрес №1: 2001:0DB0:0000:123A:0000:0000:0000:0030

Адрес №2: 2001:DB0:0:123A::30

Равны

Задание №3. В файле *orders.csv* записана (формируется студентом вручную) информация о заказах:

```
ID Клиента;ID Товара;Стоимость
1;2;200.50
2;3;99.99
1;4;123
...
12;3;200.50
```

Необходимо записать в текстовый файл *analytics.csv* информацию о клиентах: *ID Клиента;Общая сумма;Любимый товар*.

Пример:

Входной файл *orders.csv* (заголовок с названиями полей – обязателен)

```
ID Клиента;ID Товара;Стоимость
10;2;200.50
10;3;99.99
10;3;10
25;5;300
```

Выходной файл *analytics.csv* (заголовок с названиями полей – обязателен):

```
ID Клиента;Общая сумма;Любимый товар
10;310.49;3
25;300;5
```

Примечание:

1. В столбце «Стоимость» все значения > 0 и имеют не более двух знаков после запятой.
2. Любимый товар – товар, который клиент покупал чаще всего. Если таких товаров несколько, то запишите в файл любой из них.

Задание №4. Внутри запускаемого проекта (должен использоваться относительный путь) создайте директорию *example*. Внутри директории

example сгенерируйте 100 файлов размером от 1 Кб до 100 Кб (*размер каждого файла задаётся случайно*).

Задание №5. Пользователь с клавиатуры вводит два целых (*left* и *right*), положительных числа от 1 до 100. Необходимо вывести (*в консоль*) количество файлов в директории example (из задания №4), размер которых (в Кб) находится между *left* и *right* заданными пользователем (*левая и правая граница включительно*).

Задание №6. Создайте файл `article_eng.txt` и заполните его текстом любого художественного произведения на английском языке (*ограничение – размер файла должен быть не менее 10 Кб*). Необходимо определить частоту (*в долях*) повторяемости каждой латинской буквы в тексте (*остальные символы игнорировать*), отсортировать в порядке убывания частоты, результат записать в файл `article_eng_solve.txt` в формате: {буква}: {частота}.

Пример:

o: 0.095

e: 0.074

...

Примечание: символы в нижнем и верхнем регистре учитывать как один символ.