

Лабораторная работа: строки, файлы. Вариант 23

Правила выполнения лабораторной работы:

- лабораторная работа выполняется в файле lab3.py
- каждая задача решается в отдельной функции с названием taskN, где N - номер задания. Примеры: task1, task2 и т.д.
- при необходимости можно создавать дополнительные функции и обращаться к ним.
- все входные неизвестные задаются через ввод с консоли *input()* с учётом типов данных. При этом на стадиях разработки и отладки допускается задать эти параметры напрямую в программе.
- программа должна быть предназначена для работы с пользователем, то есть ввод/вывод значений сопровождается выводом текстового пояснения с помощью *print()*.
- все известные параметры задаются в качестве констант в программе.
- если в задаче указаны названия переменных, то используйте их, иначе придумайте названия сами.
- если нужно построить графики, то необходимо воспользоваться библиотекой matplotlib.
- перед выполнением заданий необходимо создать в корневом каталоге проекта папку data, в которую в дальнейшем будут записываться все файлы и директории из заданий. Для записи необходимо использовать относительные пути (../data/file_name.txt).
- в процессе выполнения собирается отчёт. Представление выполненной задачи в отчете:
 1. Постановка задачи в виде текста задания.
 2. Алгоритм решения задачи в виде блок-схемы.
 3. Текст метода решения задачи (сохранить исходный вид при копировании из IDE).
 4. Тестирование программы с различными входными параметрами (минимум 3 варианта).
 5. Проверка выполнения задач с файлами - скриншоты вывода содержимого директорий в командной строке, либо скриншот файлового менеджера.

Задание 1. Одномерный массив символов

Написать функцию `print_char_list(array)`, которая в цикле выводит в консоль элементы одномерного массива символов `array`.

Написать функцию `task1()`, в которой:

1. Задана функция, в которой создаётся одномерный массив символов из `n` элементов. Массив заполняется произвольными символами (буквами и цифрами).
2. Вызывается функция `print_char_list` для вывода полученного массива в консоль.
3. В цикле цифры заменяются символами «*» и ещё раз вызывается функция `print_char_list`.

Задание 2. Двухмерный массив символов

Написать функцию `print_char_2d_array(array)`, которая в цикле выводит в консоль элементы двумерного массива символов `array` в виде таблицы.

Написать функцию `task2()`, в которой:

1. Задана функция преобразования(`array`), в которой для входного массива `array` все односвязные последовательности символов заменяются количеством их повторений. Примеры:

$$\begin{array}{cccccccc} \cdot & * & * & * & \cdot & \cdot & 4 & 4 & 4 & \cdot \\ & & & * & & & & & 4 & \cdot \\ \cdot & \cdot & \cdot & & & & & \cdot & & \cdot \\ & * & \cdot & \cdot & \cdot & == > & 3 & \cdot & & \cdot \\ \cdot & * & * & \cdot & \cdot & & 3 & 3 & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot \end{array}$$

.	*	.	*	*	.	5	.	4	4	
.	*	.	*	*	.	5	.	1	4	
*	*	*	.	*	==>	1	5	5	.	4
.	.	*	5	.	.
*	1

! функция заключается в последовательном проходе по строкам и столбцам массива. При обнаружении вокруг символа ещё одного такого же символа, начинается внутренний цикл движения в его сторону до момента, пока вокруг все символы не будут другие

! односвязная последовательность подразумевает, что в случае когда вокруг символа есть два таких же символа, то выбирается только одно направление

2. Создаётся и случайным образом заполняется двухмерный массив символов из $(n+1) \times (n+1)$ элементов. Может быть только 2 варианта символов, например «.» и «*»
3. Вызывается для массива метод `print_char_2d_array`.
4. Вызывается функция `preobrazovanie`.
5. Для модифицированного массива вызывается функция `print_char_2d_array`.

Задание 3. Строковые переменные

Написать функцию `task3()`, в которой:

1. Выводится сообщение:

Введите текст:

2. Запрашивается ввод строковой переменной.
3. Переменная дописывается в файл, называющийся «a-words.dat», если она начинается с буквы «а», аналогично для других букв. Если слово начинается не с буквы, то запись не производится.
4. шаги 1 - 3 повторяются 5 раз.

Задание 4. Массив строк

Написать функцию `print_string_list(array)`, которая в цикле выводит в консоль элементы одномерного массива строк `array`. Написать функцию `task4()`, в которой:

1. Задан массив строк, как минимум, из 10 элементов. Каждый элемент массива представляет собой одно слово или короткую фразу из 2 - 4 слов. Слова (фразы) должны представлять собой объекты какого-либо множества (адреса сайтов, названия приложений, названия социальных сетей, термины какой-либо учебной дисциплины и т.п.).
2. Задана функция `proverka(array, string)`, которая в цикле посимвольно проверяет, содержится ли в элементе массива `array` заданная строка `string`. Если содержится, то к этому элементу добавляются два символа: пробел и символ галочки.
3. Для заданного массива вызывается функция `print_string_list`.
4. Выводится сообщение:

Введите искомый текст:

5. Запрашивается ввод строковой переменной `text_to_find`.
6. Для заданного массива и `text_to_find` вызывается функция `proverka`.
7. Для модифицированного массива вызывается функция `print_string_list`.

! Если выбрано множество сайтов, то искомым текстом может быть домен, например, выбираются сайты в домене .ru

Задание 5. Файлы запись/чтение файла

Написать функцию `task5()`, реализующую следующий функционал:

1. Выводится сообщение:

Введите текст:

2. Выполняется цикл. На каждом шаге цикла необходимо получить строку текста и записать её в файл. Цикл должен продолжаться до ввода заданного ключевого слова (задан в начале функции с помощью константы).

Задание 6. Чтение и запись с обработкой

Необходимо добавить в проект приложенный файл `variant23.txt`, содержащий результаты численного эксперимента. Написать функцию `task6()`, реализующую следующий функционал:

1. Производится чтение и анализ файла. Считываются названия (характеристики) столбцов данных (в строке, начинающейся с ключевого слова `NODE`).
2. Создаются соответствующие характеристикам одномерные массивы данных и получают их значения из файла.
3. Создаются файлы характеристик с расширением `.dat` и в каждый файл записываются значения соответствующего массива.
4. Выводятся в виде графиков значения массивов данных.

Задание 7. Создание дерева каталогов

Написать функцию `task7()`, реализующую следующий функционал:

1. Задаётся переменная с относительным путём до исходного файла (допускается реализация ввода с консоли).
2. Производится строковое чтение исходного файла и запись полученных путей в массив строк.
3. В цикле создаётся полученная структура директорий и файлов.

! Исходный для данного упражнения файл необходимо создать заранее и заполнить файловой структурой. Пример содежимого исходного файла:

```
folder1/file.txt
folder2/file1.txt
folder2/file2.txt
folder1/subfolder1/file1.dat
folder1/subfolder2/file3.txt
```

Задание 8. Мини-проект

Задание выполняется в отдельном файле `lab3_task8.py`.

Написать программу, которая выводит в терминал содержимое текущей директории и содержимое подкаталогов (подкаталогов подкаталогов и так далее) в древовидной форме.

1. Пример выполнения программы:

```
(venv) [~]: labs]$ python lab3_task8.py
├─ __init__.py
├─ lab1.py
├─ lab2.py
├─ lab3.py
├─ lab3_task8.py
├─ lab3_test1.txt
├─ lab3_test2.txt
├─ test_dir
│   └─ test_dir
│       └─ test_file2.txt
│           └─ test_file.txt.py
(venv) [~]: labs]$
```

2. Для взаимодействия с файловой системой используются методы `os.listdir`, `os.path.join`, `os.path.isdir` модуля `os`.