

Практическое задание.

Тема: Массивы и двумерные массивы.

Задание 1.

Дана таблица с указанием строки и её столбцов. Элементы таблицы имитируют папки и файлы с расширением. Создайте двумерный массив и перенесите в него все данные. После, выполните действия, написанные ниже.

| Строка № | Столбец 1 | Столбец 2 | Столбец 3 | Столбец 4 | Столбец 5 |
|----------|-------------|----------------|------------|-------------|------------|
| 1 | folder | coursework.doc | folder | pict.png | data.accdb |
| 2 | icon.ico | script.js | index.html | style.css | prog.py |
| 3 | my_song.mp3 | anapa-2003.jpg | cs_1.6.exe | folder | cheat.txt |
| 4 | notes.txt | main.py | work.pdf | cartoon.mp4 | array.py |
| 5 | project.psd | cycle.py | folder | cycle.js | turtle.py |

1. Выведете весь двумерный массив
2. Удалите все папки (folder) и замените элемент data.accdb на data.sql
выведете результат
3. Выведете все элементы, которые имеют расширение .py
Для этого, добавьте элементы в новый список и выведите его, при этом из основного списка их удалять не надо
4. По аналогии проделать ту же процедуру с файлами .js и добавить к элементу строку new_

На выходе получаем:

начальный список

```
['folder', 'coursework.doc', 'folder', 'pict.png', 'data.accdb']  
['icon.ico', 'script.js', 'index.html', 'style.css', 'prog.py']  
['my_song.mp3', 'anapa-2003.jpg', 'cs_1.6.exe', 'folder', 'cheat.txt']  
['notes.txt', 'main.py', 'work.pdf', 'cartoon.mp4', 'array.py']  
['project.psd', 'cycle.py', 'folder', 'cycle.js', 'turtle.py']
```

без папок и с заменой data

```
['coursework.doc', 'pict.png', 'data.sql']  
['icon.ico', 'script.js', 'index.html', 'style.css', 'prog.py']  
['my_song.mp3', 'anapa-2003.jpg', 'cs_1.6.exe', 'cheat.txt']  
['notes.txt', 'main.py', 'work.pdf', 'cartoon.mp4', 'array.py']  
['project.psd', 'cycle.py', 'cycle.js', 'turtle.py']
```

```
все файлы.py
prog.py main.py array.py cycle.py turtle.py

все new_файлы.js
new_script.js new_cycle.js
```

Задание 2.

Дан массив:

```
word_numb = ["ноль", "один", "два", "три", "четыре", "пять",
"шесть", "семь", "восемь", "девять"]
```

Программа просит пользователя ввести число n от 0 до 9. Если это условие выполнено, вывести элементы списка от первого элемента до n . Иначе, вывести сообщение `'Введите число <= 9'`

Пример входных/выходных данных:

```
Введите число от 0 до 9: 3
ноль
один
два
три
```

```
Process finished with exit code 0
```

В данном случае, необходимо выводить элементы списка. Цикл выстраивается в диапазоне n .

Задание 3.

Дан список:

```
bin_sy = ['11011111', '11011101', '11000111', '11011100',
'11011110']
```

С помощью цикла, вывести все элементы списка в десятичной системе счисления, а также, максимальное и минимальное число.

Перевести числа необходимо через цикл используя функцию `int`.

Для решения, можно создать ещё один список, добавить в него все десятичные числа и с помощью `max()` и `min()` вывести два числа.

Задание 4.

Дана матрица:

$$A = \begin{pmatrix} -446 & 281 & -80 \\ 465 & 432 & -122 \\ 13 & \text{error} & 8 \end{pmatrix}$$

Один из её элементов - это слово, надо это исправить и превратить его в число. Формула для превращения: слово = количество символов слова.

Создайте такую же матрицу в Python, обойдите все её элементы и их индексы циклом, с помощью **enumerate**. Далее примените **isinstance**, чтобы найти строчный тип данных. Обратитесь к индексам матрицы и замените его.

Далее, посчитайте сумму всех элементов матрицы и выведите результат.

Задание 5.

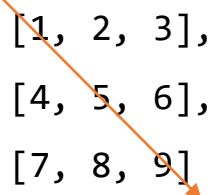
Создайте матрицу 3x3 и заполните её любыми числами. Сложите числа по диагонали справа на лево и выведите результат. Например, есть матрица:

```
matrix = [[1, 2, 3],  
          [4, 5, 6],  
          [7, 8, 9]]
```

На выходе получаем число:

```
>>> 15
```

Ответ получился при сложении: $1 + 5 + 9 = 15$



```
[1, 2, 3],  
[4, 5, 6],  
[7, 8, 9]
```

Критерии выставления оценки.

Оценка 5 – сделано 5 заданий.

Оценка 4 – сделано 4 задания.

Оценка 3 – сделано 3 задания.

Оценка 2 – сделано меньше 3 заданий.