Лабораторная работа 7

Строки

В задачах лабораторной не используйте списки и рекурсию. Условия специально указывают требуемый приём там, где он важен.

Пример: ввод нескольких чисел

Запустите ячейку и попробуйте разные значения. Здесь `split` разделяет ввод на части, `map` преобразует их в числа, а распаковка присваивает полученные значения переменным. Сохранять результат `split()` в переменную и обрабатывать его как список не требуется.

In [1]:
```
# Введите, например: 10 25
x, y = map(int, input('Введите два целых числа через пробел: ').split())
print(f'x = {x}, y = {y}')

# Введите, например: 1.5 2 3.75
a, b, c = map(float, input('Введите три вещественных числа через пробел: ').split()
print(f'a = {a}, b = {b}, c = {c}')
```

```
x = 10, y = 25
a = 1.5, b = 2.0, c = 3.75
```

Раздел 1. Операции, срезы и методы строк

Задача 1. Построение строк

Выполните два независимых подпункта.

1. Даны целое `N (> 0)` и символ `C`. Выведите строку длины `N`, состоящую из `C`.
2. Даны целое `M (> 0)` и символы `C1`, `C2`. Выведите строку длины `M`, состоящую из чередующихся символов и начинающуюся с `C1`. Значение `M` может быть нечётным.

In [ ]:
```
# 1
n = int(input('Введите N для первой строки: '))


# 2
m = int(input('Введите M для второй строки: '))
```

Задача 2. Количество гласных: цикл и методы

Дано русское предложение. Подсчитайте количество гласных букв двумя способами: сначала циклом по символам предложения, затем методом `count`. Выведите оба результата. Сравните сложность полученных подходов.

In [ ]:
```
text = input('Введите предложение: ')
vowels = 'АаЕеЁёИиОоУуЫыЭэЮюЯя'

# Способ 1: цикл по символам


# Способ 2: цикл + count
```

Задача 3. Циклический сдвиг строки

Даны непустая строка и целое неотрицательное число `K`. Выполните циклический сдвиг строки влево на `K` позиций. Например, из `abcdef` при `K = 2` получается `cdefab`. Используйте срезы; цикл не используйте.

In [ ]:
```

```

Задача 4. Сокращённая запись полного имени

Дана строка вида `Фамилия Имя Отчество`, состоящая ровно из трёх слов, разделённых одиночными пробелами. Получите строку вида `Фамилия И.О.`. Не используйте `split` и циклы.

In [ ]:
```

```

Задача 5. Имя файла и расширение

Дано имя файла, возможно, содержащее одну или несколько точек. Выведите отдельно имя без последнего расширения и расширение без точки. Если точки нет или она является последним символом, считайте, что расширение отсутствует. Используйте `rfind`.

In [ ]:
```

```

Задача 6. Маскирование номера

Дана строка длины не менее 4, содержащая номер банковской карты без пробелов. Замените все символы, кроме последних четырёх, на `*`.

In [ ]:
```

```

Задача 7. Цензор слова

Даны строка `text` и непустая подстрока `part`. Замените все непересекающиеся вхождения подстроки `part` в строке `text` на строку из звёздочек той же длины.

Подстрока может быть частью более длинной последовательности символов; границы слов учитывать не нужно. Используйте метод `replace`.

In [ ]:

Задача 8. Самое длинное слово

Дана непустая строка из слов, разделённых одним или несколькими пробелами. Найдите первое самое длинное слово. Не используйте `split`; пройдите строку циклом, накапливая текущее слово посимвольно.

In [ ]:

Задача 9. Сжатие последовательностей символов

Дана непустая строка из строчных латинских букв. Сожмите её: каждую серию одинаковых символов замените символом и количеством его повторений. Например, `aaabbc` превращается в `a3b2c1`. Используйте цикл по строке.

In [ ]:

Задача 10. Проверка имени переменной

Дана строка. Проверьте, может ли она быть упрощённым именем переменной: строка непустая, начинается с буквы или `_`, а остальные символы являются буквами, цифрами или `_`. Используйте методы `isalpha`, `isalnum` и цикл.

In [ ]:

Раздел 2. Встроенные функции для обработки символов

Задача 11. Диагностика кода

Дана строка-код, которая должна состоять только из цифр. Выведите позиции (начиная с 1) и значения всех недопустимых символов либо сообщение, что ошибок нет. Используйте `enumerate`: здесь одновременно нужны символ и его позиция.

In [ ]:

Задача 12. Сравнение двух версий текста

Даны две строки, возможно разной длины. Выведите позиции (начиная с 1), на которых символы различаются в их общей части, и количество таких позиций. Используйте `zip`: он сразу образует пары символов и заканчивается вместе с более короткой строкой, поэтому индексы и `min` не нужны.

In [ ]:

Задача 13. Фрагмент по введённым позициям

Дана строка и два целых индекса `start`, `end`, введённые в одной строке через пробел. Получите набор символов от `start` (включительно) до `end` (невключительно). Для ввода индексов обязательно используйте `map`.

In [ ]:

Задача 14. Фильтрация символов

Дана строка. Используя `filter` и `lambda`, получите две новые строки: одну только из цифр исходной строки, вторую — только из знаков препинания `.`, `,`, `!`, `?`, `;`, `:`.

In [ ]:

Задача 15. Проверка пароля

Дана строка-пароль. Пароль допустим, если его длина не менее 8, все символы — буквы или цифры, и в нём есть хотя бы одна строчная буква, заглавная буква и цифра. Используйте `all`, `any` и `map`.

In [ ]: