



работа на уроке (22.10)

1. `range(start, stop, step)` `step = -1`

Программа принимает на вход число `x`. Сформировать два списка длины `2x-1`:

- Первый — содержащий все элементы от `0` до `x` и от `x` до `0` подряд (`x` не повторяется). Результат сохранить в переменную `up_n_down`.
- Второй — содержащий все элементы от `x` до `0` и от `0` до `x` подряд (`0` не повторяется). Результат сохранить в переменную `down_n_up`.

Вывести обе переменные

Ввод	Вывод
7	[0, 1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1, 0] [7, 6, 5, 4, 3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7]

2. `len()` `range()` `.extend()` `.append()`

Даны два списка. Слить их через один (сначала элемент из первого списка, затем элемент из второго и т.д.) и вывести. Если один из списков длиннее, то в конце добавить “остаток”.

NB! Мы не знаем, какой из списков длиннее.

Ввод	Вывод
[1, 2, 3, 4, 5, 6] ['a', 'b', 'c', 'd', 'e', 'f']	[1, 'a', 2, 'b', 3, 'c', 4, 'd', 5, 'e', 6, 'f']
[1, 2, 3] ['a', 'b', 'c', 'd', 'e', 'f']	[1, 'a', 2, 'b', 3, 'c', 'd', 'e', 'f']
[1, 2, 3, 4, 5, 6] ['a', 'b', 'c', 'd']	[1, 'a', 2, 'b', 3, 'c', 4, 'd', 5, 6]

3. `type()` `list[start:stop]`

Известно, что в списке `t` сначала идут *числовые элементы*, а затем *строковые*. Напишите программу, которая принимала бы на вход один список и делила его на два по типам данных, печатая оба.

Ввод	Вывод
[0, 9, 56, -34, 5, 'a', 'abc', 'def']	[0, 9, 56, -34, 5] ['a', 'abc', 'def']

3. `len()` `range()` `in`

Дан список элементов, который содержит `n` различных чисел из отрезка `[0, n]`.
Напишите код, который принимает на вход список и печатает недостающее число из этого списка.

Ввод	Вывод
<code>[0, 2]</code>	1
<code>[1, 2, 3]</code>	0

4. `while-loop` `.split()` `.append()`

Напишите программу, которая будет считать стоимость «корзины», набранной в онлайн-магазине.

Каждый раз с новой строки будет вводиться стоимость товара и его количество, разделённые пробелом, пока не будет введена пустая строка. В качестве вывода нужно представить список «подытогов» по каждому товару (**NB!** вывести их нужно уже после всех вводов), каждый с новой строки, а затем — сумму этих подытогов.

Ввод	Вывод
1000 1 300 5 150 2	1000 1500 300 Сумма: 2800
10000 10	100000 Сумма: 100000

5. `.split()` `.isupper()` `list[index]` `string.punctuation`

С клавиатуры вводится предложение. Найти в нём все имена (пока будем считать, что любое слово не в начале предложения, написанное с большой буквы — это имя) и заменить их на строку `NAME`.

NB! Обратите внимание, что знаки препинания после измененных слов сохраняются.

Ввод	Вывод
Вслед за Элен перешла и маленькая княгиня от чайного стола.	Вслед за NAME перешла и маленькая княгиня от чайного стола.
Вскоре после приезда Ростовых в Петербург Берг сделал предложение Вере, и предложение его было принято.	Вскоре после приезда NAME в NAME NAME сделал предложение NAME, и предложение его было принято.
В Финляндской войне ему удалось также отличиться.	В NAME войне ему удалось также отличиться.

6. `string[-index]` `.append()` `.sort()`

Вводятся грузинские слова, записанные в латинской транскрипции. Ввод останавливается, когда пользователь введёт пустую строку.

Определите, сколько среди введённых слов — существительные во множественном числе (они имеют показатель *-eb-*, который вставляется перед окончанием слова, которое всегда занимает ровно одну букву). Выведите их по одному на строку в **алфавитном порядке**.

Ввод	Вывод
knuti knute <u>b</u> i kate <u>b</u> i gzaghli gzagh <u>l</u> ebi mimzidveli ge <u>b</u> u ze <u>b</u> ra	gebu gzaghlebi katebi knutebi