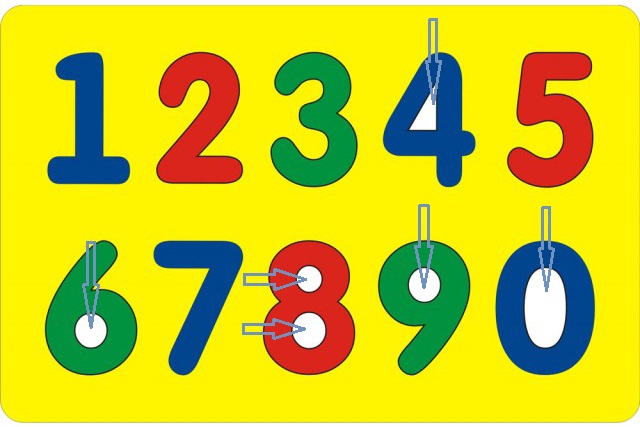
**Задание 1**

Создать функцию func(var\_a, var\_b), которая анализирует переменные (var\_a и var\_b могут быть типа "str" или "int") и в зависимости от значения переменных возвращает один из следующих результатов:  
- "str" - если хотя бы одна из переменных является строкой;  
- ">" - если var\_a больше var\_b;  
- "=" - если значения переменных равны;  
- "<" - если var\_a меньше var\_b.

**Пример**  
Вызов функции: func(5, '2')  
Возвращает: 'str'

**Задание 2**

Создать функцию count\_holes(value), которая принимает аргумент value - целое число или строка содержащая целое число.  
Функция возвращает целое число - количество "отверстий" (смотри рисунок) в записи этого числа, или строку 'error', если переданый аргумент не удовлетворяет требованиям: число не целое или вообще не является числом. Нули в начале записи числа не учитывать, если таковые имеются.



**Пример**  
Вызов функции: count\_holes('08824')  
Возвращает: 5

**Задание 3**

Создать функцию hangman(word, letters), которая принимает два аргумента. word - строка и letters - список букв.  
Функция заменяет буквы в строке word на символ подчеркивания \_, если их нет в списке letters и возвращает измененную строку.

Для читаемости полученой строки:  
- добавте по одному пробелу справа и слева символа подчеркивания;

- между двумя символами подчеркивания должен быть один пробел;  
- в начале и в конце строки пробелов не должно быть.

**Пример**  
Вызов функции: hangman('python', ['a', 'r', 'y', 'i', 'o'])  
Возвращает: '\_ y \_ \_ o \_'

**Задание 4**

Создать функцию double\_tuple(value), которая принимает кортеж любых чисел или строк и модифицирует его в кортеж кортежей по два элемента (парами).

**Пример**  
Вызов функции: double\_tuple((1,4,8,6,3,7,1))  
Возвращает: ((1,4),(8,6),(3,7),(1,))

**Задание 5**

Дан текст text и ограничение длины текста limit (в количестве символов). Необходимо, создать функцию trimmed\_text(text, limit), которая в случае, если текст не помещается в ограничение обрезает его, но при этом слова не должны обрываться на середине (исключение первое слово), и в конце нужно добавить троеточие ("..."). Перед троеточием не должно быть других разделительных знаков

**Пример**  
Вызов функции: trimmed\_text("Proin eget tortor risus.", 24)  
Возвращает: "Proin eget tortor risus."  
  
Вызов функции: trimmed\_text("Proin eget tortor risus.", 23)  
Возвращает: "Proin eget tortor..."  
  
Вызов функции: trimmed\_text("Proin eget tortor risus.", 6)  
Возвращает: "Pro..."

**Задание 6**

Дан текст text, который может содержать буквы латинского алфавита, пробелы и знаки препинания , . : ; ! ?. Необходимо, создать функцию find\_most\_frequent(text), которая возвращает список слов в нижнем регистре, наиболее часто встречающиеся в тексте, в алфавитном порядке.

**Примечание** регистр в словах не имеет значения слова "hello" и "Hello" считаются одним словом.

**Пример**  
Вызов функции: find\_most\_frequent('Hello, hello, my dear!')  
Возвращает: ['hello']  
  
Вызов функции: find\_most\_frequent('to understand recursion you need first to understand recursion...')  
Возвращает: ['recursion', 'to', 'understand']