ГОСТ 19.202-78 Единая система программной документации. Спецификация. Требования к содержанию и оформлению. Настоящий стандарт устанавливает форму и порядок составления программного документа "Спецификация". Спецификация является основным программным документом для компонентов, применяемых самостоятельно, и для комплексов.

a=list(input("Введите числа"))

for i in range(len(a)):

print("Цифра", a[i], "встречается", a.count(a[i]), "раз(а)")

a=list(input("Введите первый список чисел "))

b=list(input("Введите первый список чисел "))

c = list(set(a) & set(b))

for i in c:

print("Число:",i,"встречается в двух списках",a.count(i)+b.count(i),"раз(а)")

s='''Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated'''

print('''Кол-во слов: ''', len(s.split()))

lang = []

union = set()

all = set()

for i in range(int(input())):

m = int(input())

a = {input() for j in range(m)}

all.update(a)

if i == 1:

union.update(a)

else:

union &= a

print(len(union))

print('\n'.join(sorted(union)))

print(len(all))

print('\n'.join(sorted(all)))

a={1,2,3,4,5,6,7}

b={1,2,3,4,5,6,7,8,9,10,11}

c= b.difference(a)

print(c)

text='Дан текст состоящий из количества строк текста и самого текста Вывести на экран построчно каждое слово и сколько раз оно встречается в тексте Для реализации используйте словари'

text=text.split()

a={}

for i in text:

a[i] = text.count(i)

for key,value in a.items():

print(key, ':', value)

b='''

clown

circus

xqc

trainwreckTV

'''

a=b.split()

c={i: len(i) for i in a}

key\_val = c.items()

key\_val\_list = list(key\_val)

kv = max(key\_val\_list, key=lambda i : i[1])

print("Самое длинное слово:",kv[0], "Букв:",kv[1])

text = '''

Напишите программу, которая считывает текст из файла (в файле может быть

больше одной строки) и выводит самое частое слово в этом тексте и через

пробел то, сколько раз оно встретилось. Если таких слов несколько, вывести

лексикографически первое (можно использовать оператор < для строк).

В качестве ответа укажите вывод программы, а не саму программу.

Слова, написанные в разных регистрах, считаются одинаковыми.

'''

lst\_no = ['.', ',', ':', '!', '"', "'", '[', ']', '-', '—', '(', ')', '?', '\_', '`' ]

lst = []

for word in text.lower().split():

if not word in lst\_no:

\_word = word

if word[-1] in lst\_no:

\_word = \_word[:-1]

if word[0] in lst\_no:

\_word = \_word[1:]

lst.append(\_word)

\_dict = dict()

for word in lst:

\_dict[word] = \_dict.get(word, 0) + 1

# сортируем словарь посредством формирования списка (значение, ключ)

\_list = []

for key, value in \_dict.items():

\_list.append((value, key))

\_list.sort(reverse=True)

# самое частое слово в этом тексте

print(f'самое частое слово в этом тексте -> `{\_list[0][1]}`, использовалось `{\_list[0][0]}` раз.')

from collections import defaultdict

from sys import stdin

clients = defaultdict(lambda: defaultdict(int)) #O(n)

for line in stdin.readlines(): # O(n)

client, thing, value = line.split()

clients[client][thing] += int(value)

for client in sorted(clients): # O(n^2)

print(client + ':')

for thing in sorted(clients[client]):

print(thing, clients[client][thing])

#Сложность алгоритма - O(n) \* O(n) \* O(n^2) = O(n^4)

a={'Архангельская область1' : 'Архангельск' , 'Архангель1ская область' :'Новодвинск',

'Архангельская об1ласть' : 'Северодвинск', 'Архангельская1 область' :'Шенкурск', 'Арханге1льская область' :'Котласс'}

l={'Л1енинградская область': 'Санкт-Петербург', 'Ленинградс1кая область':'Пушкин', 'Ленингра1дская область':'Павловск'}

val1,val2,val3= 'Архангельск' , 'Пушкин', 'Котласс'

print('Архангельская область' if (val1 in a.values()) == True else 'Ленинградская область')

print('Архангельская область' if (val2 in a.values()) == True else 'Ленинградская область')

print('Архангельская область' if (val3 in a.values()) == True else 'Ленинградская область')