**Практическая работа №7. Функции. Область видимости переменных**

**Шумов Влад**

**ИСП23.1А**

https://github.com/VladKristoff/PR7-Functions

**Вариант 8, задача 1:**

**Формат ввода**

Вводится строка фраз, разделенных точкой с запятой и пробелом. Затем вводится строка-образец.

**Формат вывода**

Вывести через знак #, окруженный пробелами, из исходной строки только те фразы, у которых не больше 3 общих символов со строкой-образцом. Повторяющиеся символы считать за один, регистр не важен.

**Пример 1**

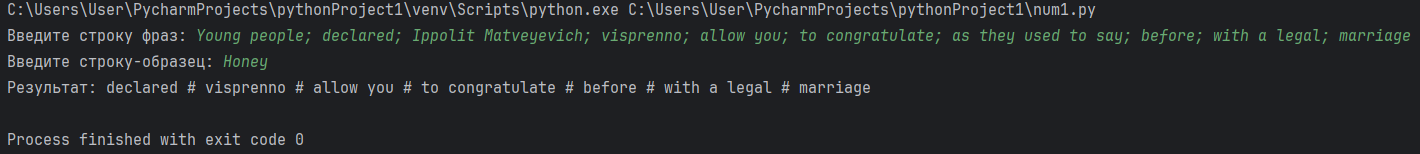
| Ввод |
| --- |
| Young people; declared; Ippolit Matveyevich; visprenno; allow you; to congratulate; as they used to say;  before; with a legal; marriage  Honey |
| Вывод |
| declared # visprenno # allow you # to congratulate # before # with a legal # marriage |

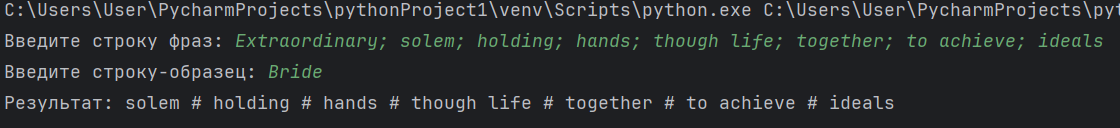
**Пример 2**

| Ввод | Вывод |
| --- | --- |
| Extraordinary; solemn; holding hands; through life; together; to achieve; ideals  Bride |  |

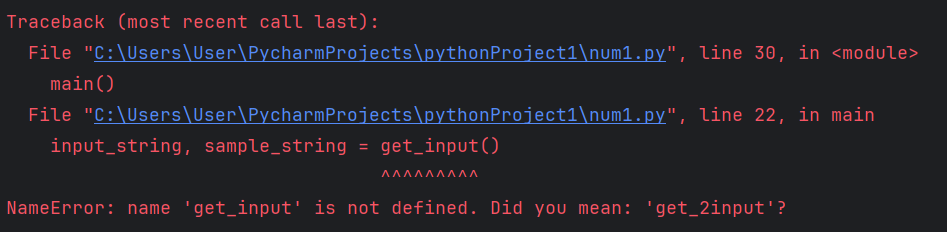
def get\_input():  
 input\_string = input("Введите строку фраз: ")  
 sample\_string = input("Введите строку-образец: ")  
 return input\_string, sample\_string  
  
def split\_phrases(input\_string):  
 return input\_string.split('; ')  
  
def get\_unique\_chars(string):  
 return set(string.lower())  
  
def filter\_phrases(phrases, sample\_set):  
 filtered\_phrases = []  
 for phrase in phrases:  
 phrase\_set = get\_unique\_chars(phrase)  
 common\_chars = len(sample\_set.intersection(phrase\_set))  
 if common\_chars <= 3:  
 filtered\_phrases.append(phrase)  
 return filtered\_phrases  
  
def main():  
 input\_string, sample\_string = get\_input()  
 phrases = split\_phrases(input\_string)  
 sample\_set = get\_unique\_chars(sample\_string)  
 filtered\_phrases = filter\_phrases(phrases, sample\_set)  
 result = ' # '.join(filtered\_phrases)  
 print(f"Результат: {result}")  
  
  
main()

**Тестирования**

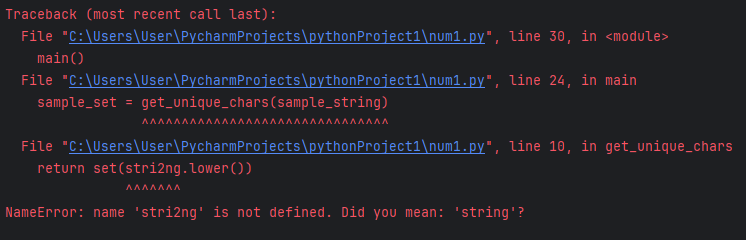




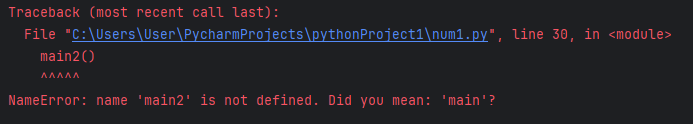
**Ошибки:**



Неправильное название функции



Неправильное имя переменной



Неправильный вызов функции

**Вариант 7, Задача 2**

**Формат ввода**

Вводятся три строки слов, записанных через пробел.

**Формат вывода**

Через плюс без повторений в произвольном порядке вывести все слова, которые встречались хотя бы в одной строке.

Затем вывести самое длинное из выбранных слов. Если таких несколько, самое маленькое при лексикографическом сравнении.

**Пример 1**

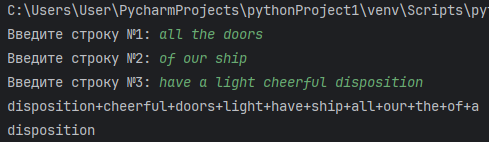
| Ввод | Вывод |
| --- | --- |
| all the doors  of our ship  have a light cheerful disposition | all+ship+have+doors+the+disposition+a+our+light+cheerful+of  disposition |

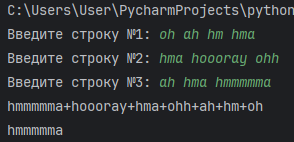
**Пример 2**

| Ввод | Вывод |
| --- | --- |
| oh ah hm hma  hma hoooray ohh  ah hma hmmmmma | ohh+hmmmmma+hma+hoooray+hm+ah+oh  hmmmmma |

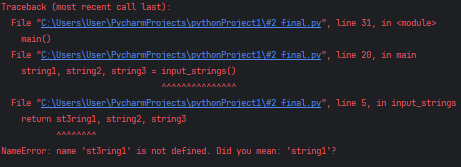
def input\_strings():  
 string1 = input('Введите строку №1: ')  
 string2 = input('Введите строку №2: ')  
 string3 = input('Введите строку №3: ')  
 return string1, string2, string3  
  
  
def process\_strings(string1, string2, string3):  
 words\_set = set(string1.split() + string2.split() + string3.split())  
 words\_list = list(words\_set)  
 return words\_list  
  
  
def find\_longest\_word(words):  
 words.sort(key=lambda x: (-len(x), x))  
 return words[0] if words else ""  
  
  
def main():  
 string1, string2, string3 = input\_strings()  
  
 unique\_words = process\_strings(string1, string2, string3)  
  
 longest\_word = find\_longest\_word(unique\_words)  
  
 print('+'.join(unique\_words))   
 print(longest\_word)   
  
  
  
main()

**Тестирования**

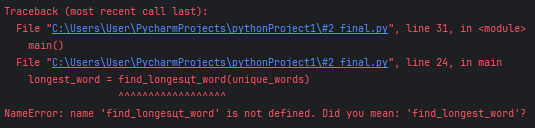




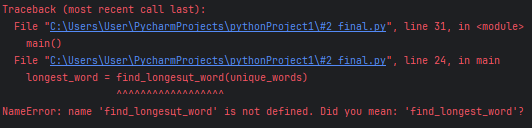
**Ошибки:**



Возвращается переменная с неправильным названием



Неправильное название функции



Неправильный ввод