

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Введение в информационные технологии»**  
**Тема: Основные управляющие конструкции языка Python**

Студент гр. 9304

\_\_\_\_\_

Мохаммед. А.А

Преподаватель

\_\_\_\_\_

Размочаева Н.В

Санкт-Петербург

2019

## Цель работы.

Научиться использовать основные управляющие конструкции языка Python.

## Задание.

С помощью новых инструментов —модуля `wikipedia` и модуля `help_wiki_function` —напишите программу, которая принимает на вход строку вида

название\_страницы\_1,                    название\_страницы\_2,  
...название\_страницы\_n, сокращенная\_форма\_языка

и делает следующее:

1. Проверяет, есть ли такой язык в возможных языках сервиса, если нет, выводит строку "no results" и завершает выполнение программы. В случае, если язык есть, устанавливает его как язык запросов в текущей программе.

2. Ищет максимальное число слов в кратком содержании страниц "название\_страницы\_1", "название\_страницы\_2", ... "название\_страницы\_n", выводит на экран это максимальное количество и название страницы (т.е. `title`), у которой оно обнаружилось. Считается, что слова разделены пробелами и символами.

Если максимальных значений несколько, выведите последнее.

3. Строит список-цепочку из страниц и выводит полученный список на экран.

Элементы списка-цепочки - это страницы "название\_страницы\_1", "название\_страницы\_2", ... "название\_страницы\_n", между которыми может быть одна промежуточная страница или не быть промежуточных страниц.

Предположим, нам на вход поступила строка:

Айсберг, IBM, ru

В числе ссылок страницы с названием "Айсберг", есть страница с названием , которая содержит ссылку на страницу с названием "Буря", у которой есть ссылка на страницу с названием "IBM" -- это и есть цепочка с промежуточным звеном в виде страницы "Буря".

Гарантируется, что существует или одна промежуточная страница или ноль: т.е. в числе ссылок первой страницы можно обнаружить вторую.

Цепочка должна быть кратчайшей, т.е. если существуют две цепочки, одна из которых содержит промежуточную страницу, а вторая нет, стройте цепочку без промежуточного элемента.

Пример входных данных:

Айсберг, IBM, ru

Пример вывода:

```
115 IBM
[ 'Айсберг', 'Буря', 'IBM' ]
```

Первая строка содержит решение подзадачи №2, вторая - №3.

### **Основные теоретические положения.**

Для определения циклов с известным количеством итераций используется конструкция for. Синтаксис цикла for следующий:

```

for <переменная> in <коллекция>:
    <действие 1>
else: #необязательный блок
    <действие 2>

```

Операция `in` — операция проверки на вхождение. Не все типы данных могут быть итерируемыми, т.е. переменные не всех типов данных могут состоять из элементов, которые можно было бы обойти в цикле.

Когда требуется перебрать не элементы последовательности, а ее индексы, можно использовать функцию `range()`.

Функция `range()` принимает следующие параметры: `range (start, stop[, step])` где `start` - это начальное значение, `stop` - значение, до которого необходимо сгенерировать последовательность, `step` - шаг (по умолчанию равен единице). Все аргументы функции `range()` должны быть целыми числами.

Оператор `if` позволяет реализовать ветвление программы. Его синтаксис:

```

if <условие 1>:
    <действие 1> # Данный блок сработает, если <условие 1> истинно
elif <условие 2>: # Необязательный блок 1 – ложь, 2 – истина
    <действие 2>
else: # Необязательный блок
    <действие 3> # Данный блок сработает, если ложны оба условия

```

Данная конструкция проверяет истинность набора условий и в зависимости от того, какие из них выполняются, переводит управление к соответствующим блокам кода. Условием может быть любое выражение, которое можно привести к типу `bool`.

Файл с программным кодом в языке Python называется модуль. Вы можете использовать существующие модули в своих программах, для этого

вам необходимо их импортировать. Для примера рассмотрим модуль `math`, который содержит имена (т.е. переменные и функции) для работы с числами:

`>>> import math` После импорта модуля вы можете использовать все имена, которые в нем определены, используя следующий синтаксис: `<имя_модуля>.<имя>`

### Выполнение работы.

Разработанный алгоритм решает поставленную задачу следующим образом:

1) Считывает строку и разбивает её на список строк с помощью метода `строка.split()`;

2) Если последний элемент списка не является ключом к какому-либо значению из словаря `wikipedia.languages()` из модуля `wikipedia`, то выводит строку «no results» и завершает работу;

3) В противном случае устанавливает найденный язык как язык запросов в текущей системе и удаляет элемент списка, обозначающий язык;

4) Находит максимальную длину списка, полученного разбиением `page.summary` текущей страницы на список строк, выводит это количество и `title` страницы;

5) В цикле от первого до предпоследнего элемента списка делает следующее: добавляет текущий элемент в список-цепочку; если следующего элемента нет в списке ссылок с текущей страницы, проверяет ссылки, ведущие со страницы: на каждой ищет ссылку, которая ведёт на следующую страничку из заданного списка. Когда ссылка найдена, добавляет её в список-цепочку и прерывает цикл;

6) Выводит полученный список-цепочку на экран

Полное рабочее решение см. в Приложении А.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	Айсберг, IBM, ru	115 IBM ['Айсберг', 'Буран', 'IBM']	Sample input & output Результат образцом
2	Буран, IBM, ru	169 Буран (космический корабль) ['Буран', 'IBM']	Получен ож идаемый результат
3	Чуумпу далай, ХотуАмерика,Конт инент,sah	77 Чуумпу далай ['Чуумпу далай', 'Америкалар', 'ХотуАмерик а', 'Континент']	Получен ож идаемый результат
4	One, two, three	no results	Получен ож идаемый результат
5	A quick brown fox jumps over the lazy dog.	no results	Получен ож идаемый результат

## Выводы.

Были изучены основные управляющие конструкции языка Python, было освоено применение циклов, условий, модулей.

Разработана программа, работающая с интернет-страницами сервиса wikipedia: подсчитывающая количество слов в кратком содержании страниц, строящая список-цепочку из страниц. Для работы использовались модули wikipedia и help\_wiki\_function, условия, цикл for, методы класса str.

## ПРИЛОЖЕНИЕ

### АИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
import wikipedia

import help_wiki_function

def find_max_title(lst):
    maximum = 0
    for current_page in lst:

        if len(wikipedia.page(current_page).summary.split()) >= maximum:

            maximum = len(wikipedia.page(current_page).summary.split())
            title = wikipedia.page(current_page).title

    return(maximum, title)

def create_chain(pages):
    answer = []

    for i in range(len(pages)-1):

        answer.append(pages[i])

        if pages[i+1] not in wikipedia.page(pages[i]).links:

            for checked_outer_link in wikipedia.page(pages[i]).links:

                if help_wiki_function.is_page_valid(checked_outer_link):

                    if pages[i+1] in wikipedia.page(checked_outer_link).links:

                        answer.append(checked_outer_link)

                    break
```



```
        answer.append(pages[-1])
    return(answer)

pages = input().split(' ', ' ')

if pages[-1] in wikipedia.languages():

    wikipedia.set_lang(pages[-1])
    pages.pop()

    print(find_max_title(pages)[0], find_max_title(pages)[1])

    print(create_chain(pages))
else:
    print("no results")
```