

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Информатика»
Тема: Основные управляющие конструкции. WikipediaAPI

Студент гр. 0382

Ильин Д.А.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Изучить основные управляющие конструкции языка Python и его модуль WikipediaAPI.

Задание.

Используя вышеописанные инструменты, 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. её title), у которой оно обнаружилось. Считается, что слова разделены пробельными 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 "Буран", у которой есть ссылка на страницу с названием "IBM" -- это и есть цепочка с промежуточным звеном в виде страницы "Буран".

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', 'Буран', 'IBM']

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

Ход работы:

В начале программы импортируется модуль wikipedia.

Далее в `a` считывается строка и преобразуется в массив, посредством `split(' ', '')`.

В переменную `b` передаётся последний элемент(введённое укороченное название языка, по которому будут происходить запросы) `a`, который потом удаляется из массива.

Потом выполняется проверка: есть ли такой язык в списке языков википедии, если нет, то на экран выводится: "no results", введённый язык становится языком запросов, а также вызывается функция `Max`, аргументом которой является массив `a` и `mass`, аргументами которой являются:

- 1) Массив `a`
- 2) Массив, который должен получиться
- 3) Индекс элемента, до которого уже дошла функция `mass`(при вызове он 0)

В функции `Max` создаются переменные `kol` и `rez`, первая из них содержит максимальное кол-во слов краткого содержания введённых страниц википедии, а во второй название страницы с максимальным кол-ом слов. При помощи цикла `for` находятся `kol` и `rez`, соответствующие введённым данным.

В функции `mass` используется рекурсия. Сначала в список `kon` добавляется элемент с индексом номера от которого вызвана функция массива `a`. Из поданного массива и нынешнего номера страницы, из массива

а, функции(который подаётся в массив), ищется следующая страница сначала прямо, а потом через одну дополнительную ссылку (в таком случае добавляет её в массив), когда такая нашлась вызывается снова эта функция от тех же значений, за исключением номера, который увеличивается на 1.

Так происходит пока номер не становится равным последнему номеру элемента в массиве а, как только так происходит, в список `kon` добавляется последний элемент из массива а, а затем выводится сам список `kon`.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Айсберг, Атмосфера, Иттрий, ru	119 Атмосфера ['Айсберг', 'Абляция (гляциология)', 'Атмосфера', 'Азот', 'Иттрий']	Программа работает верно
2.	Айсберг, Атмосфера, Иттрий, rrrr	no results	Такого языка в википедии нет
3.	Айсберг, ru	73 Айсберг ['Айсберг']	Программа работает верно
4.	Атмосфера, Азот, ru	119 Атмосфера ['Атмосфера', 'Азот']	Программа работает верно

Выводы.

Были изучены основы языка программирования Python3, а также его модуля Wikipedia.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb1.py

```
import wikipedia

a = input().split(' ', ' ')
b = a[len(a) - 1]
a.pop(len(a) - 1)
kon = []

def Max(a):
    kol = 0
    rez = 0
    for i in a:
        t =
wikipedia.page(i)

        if kol <=
len(t.summary.split()):

            kol =
len(t.summary.split())

            rez =
wikipedia.page(i).title

    print(kol, rez)

def is_page_valid(page):
    try:

        wikipedia.page(page)

    except Exception:

        return False
```



```

        return True

def mass(a, kon, nach):

    #print(1)

    if nach < len(a) -
1:

        t =
wikipedia.page(a[nach]).link
s

    kon.append(a[nach])

    if a[nach + 1]
in t:

        #print(kon)

        mass(a, kon,
nach + 1)

    else:

        for i in t:

            #print(i)

            if
is_page_valid(i):

                l =
wikipedia.page(i).links

                if
a[nach + 1] in l:

                    kon.append(i)

            #print(kon)

            mass(a, kon, nach + 1)

        break

```

```
        else:

kon.append(a[nach])

        print(kon)

if b in
wikipedia.languages():

    wikipedia.set_lang(b)

    Max(a)

    mass(a, kon, 0)

else:

    print("no results")
```