

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение

Дисциплина «Информатика»

Отчёт по лабораторной работе №3

Регулярные выражения

Вариант №466513

Выполнил

Линейский Аким Евгеньевич

P3115

Проверил

Белокон Юлия Алексеевна

Санкт - Петербург 2024

# Содержание

[Содержание 2](#_Toc180181749)

[Задание 3](#_Toc180181750)

[Основное задание 3](#_Toc180181751)

[Дополнительное задание №1. 4](#_Toc180181752)

[Дополнительное задание №2 7](#_Toc180181753)

# Задание

# Основное задание

1. Реализовать программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице (таблица 1.). 2. Для своей программы придумайте минимум 5 тестов. Каждый тест является отдельной сущностью, передаваемой регулярному выражению для обработки. Для каждого теста необходимо самостоятельно (без использования регулярных выражений) найти правильный ответ. После чего сравнить ответ, выданный программой, и полученный самостоятельно. Все 5 тестов необходимо показать при защите.

3. Программа должна считать число смайликов определённого вида (вид смайлика описан в таблице вариантов) в предложенном тексте. Все смайлики имеют такую структуру:

[глаза][нос][рот].

Вариантом является различные наборы глаз, носов и ртов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер в ИСУ % 6 | Глаза | Номер в ИСУ % 4 | Нос | Номер в ИСУ % 8 | Рот |
| 0 | 8 | 0 | - | 0 | ( |
| 1 | ; | 1 | < | 1 | ) |
| 2 | X | 2 | -{ | 2 | P |
| 3 | : | 3 | <{ | 3 | | |
| 4 | = |  |  | 4 | \ |
| 5 | [ |  |  | 5 | / |
|  |  |  |  | 6 | O |
|  |  |  |  | 7 | = |

(таблица 1.)

**Решение:**

Номер ИСУ: 466513.

466513%6=1. Глаза: ";"

466513%4=1. Ноc: "<"

466513%8=1. Рот: ")"

Смайлик: ";<)"

Тесты:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Вывод | Результат программы |
| 1 | 123;a()</k&@; | 0 | 0 |
| 2 | ;a<;<)/;>/ | 1 | 1 |
| 3 | Abos;<))//.>//as2>/;/>;<)( | 2 | 2 |
| 4 | ;;)<);<):<));<<;<);;<));<) | 4 | 4 |
| 5 | :0><\(;<);>;>;(<),>::.;<):.:<);<));:<):0><\(;<);>;>;(<),>:<):.:<);<));:<) | 5 | 5 |
| 6 | :;-</)0|.P]x\*;<)({/\?-}{-X8O|;<)\)(=[;>-x | 2 | 2 |
| 7 | :-(:-\_:-0:0|;<)\:0P8<{|[\0;</;<);</8<{0=-O:-(:-\_:-0:0|;<)\:0P8<{|[\0;</;<);</8<{0=-O | 4 | 4 |

Исходный код программы:

import re; print(len(re.findall(r';<\)', input())))

# Дополнительное задание №1.

1. Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице. (466513%6=1).

Довольно распространённая ошибка ошибка – это повтор слова. Вот в предыдущем предложении такая допущена. Необходимо исправить каждый такой повтор. Повтор это – слово, один или несколько пробельных символов, и снова то же слово.

Пример:

|  |  |
| --- | --- |
| Ввод | Вывод |
| Довольно распространённая ошибка ошибка – это лишний повтор повтор слова слова. Смешно, не не правда ли? Не нужно портить хор хоровод. | Довольно распространённая ошибка – это лишний повтор слова. Смешно, не правда ли? Не нужно портить хор хоровод. |

**Решение:**

Тесты:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Вывод | Результат работы программы |
| 1 | Довольно распространённая ошибка ошибка – это лишний повтор повтор слова слова. Смешно, не не правда ли? Не нужно портить хор хоровод. | Довольно распространённая ошибка – это лишний повтор слова слова. Смешно, не правда ли? Не нужно портить хор хоровод. | Довольно распространённая ошибка – это лишний повтор слова. Смешно, не правда ли? Не нужно портить хор хоровод |
| 2 | Message without dublicates | Message without dublicates | Message without dublicates |
| 3 | hello hello, world! | hello, world! | hello, world! |
| 4 | Lorem ipsum        ipsum dolor sit amet amet, consectetur adipiscing adipiscing elit. Praesent eget fringilla urna. Nulla mollis porttitor dapibus. Mauris neque neque sem sem, ullamcorper ullamcorper vitae turpis nec, pulvinar bibendum sem. Donec suscipit suscipit tempor dui, eget eget tincidunt nisi nisi viverra vitae. Duis; nec nec dapibus magna, nec ! nec finibus risus. Aliquam - aliquam quis quis tincidunt tellus. Ut imperdiet, metus id. | Lorem ipsum dolor sit amet, consectetur adipiscing eIit. Praesent eget fringilla urna. Nulla m011is porttitor dapibus. Mauris neque sem, ullamcorper vitae turpis пес, pulvinar bibendum sem. Donec suscipit tempor dui, eget tincidunt nisi viverra vitae. Duis; пес dapibus тадпа, пес fnibus risus. AIiquam - aIiquam quis tincidunt tellus. Ut im erdiet, metus id. | Lorem ipsum dolor sit amet, consectetur adipiscing eIit. Praesent eget fringilla urna. Nulla m011is porttitor dapibus. Mauris neque sem, ullamcorper vitae turpis пес, pulvinar bibendum sem. Donec suscipit tempor dui, eget tincidunt nisi viverra vitae. Duis; пес dapibus тадпа, пес fnibus risus. AIiquam - aIiquam quis tincidunt tellus. Ut im erdiet, metus id. |
| 5 | gIT . Git ... hub ! ; git a ; hub . hub hub     git hub ! git hub & hub git git git hub hub git | gIT . Git ... hub ! ; git a ; hub hub     git hub ! git hub git git hub git | gIT . Git ... hub ! ; git a ; hub hub     git hub ! git hub git git hub git |
| 6 | Push push push !!! push request - request out \_\_\_ error, error! | Push push !!! push request out \_\_\_ error! | Push push !!! push request out \_\_\_ error! |

Исходный код программы:

**import** re

inp = **input**()

words = re.split(r"(\W+)", inp)

words1 = []

i=0

**while** i<**len**(words)-2:

words1.append(words[i])

**if** words[i]==words[i+2] **and** re.match('\w',words[i]) **is** **not** None:

words1.append(words[i+1])

i+=2

i+=1

words1.append(words[-2])

words1.append(words[-1])

**while** "" **in** words1:

words1.remove("")

i =0

**while** i<**len**(words1)-1:

**if** re.match('\W',words1[i]) **is** **not** None **and** re.match('\W',words1[i+1]) **is** **not** None:

words1.pop(i)

**else**: i+=1

**print**(''.join(words1))

# Дополнительное задание №2

1. Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице. (466513%8=1).

С помощью регулярного выражения найти в тексте слова, в которых встречается строго одна гласная буква (встречаться она может несколько раз). Пример таких слов: окно, трава, молоко, etc. После чего данные слова требуется отсортировать сначала по увеличению длины слова, а затем лексикографически.

Пример:

|  |  |
| --- | --- |
| Ввод | Вывод |
| Классное слово – обороноспособность, которое должно идти после слов: трава и молоко. | и  идти  слов  слово  трава  должно  молоко  обороноспособность |

**Решение:**

Тесты:

|  |  |  |
| --- | --- | --- |
| Ввод | Вывод | Результат работы программы |
| Классное слово – обороноспособность, которое должно идти после слов: трава и молоко. | и  идти  слов  слово  трава  должно  молоко  обороноспособность | и  идти  слов  слово  трава  должно  молоко  обороноспособность |
| а и у о е | а  е  и  о  у | а  е  и  о  у |
| Молоко1 молоко2 молоко3 | молоко1  молоко2  молоко3 | молоко1  молоко2  молоко3 |
| молоко астра | астра  молоко | астра  молоко |
| молоко молокоа молокоб молоков молокоаа молокобб молоковв | молоко  молокоб  молоков  молокобб  молоковв | молоко  молокоб  молоков  молокобб  молоковв |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus elit arcu, ullamcorper ut nisi sit amet, gravida auctor lectus. Cras eget dictum nunc, sit amet egestas dui. In non commodo magna. Sed non aliquam mi, ut semper urna. In et dolor dapibus, tristique massa et, feugiat urna. Maecenas id orci et arcu hendrerit varius. Phasellus sollicitudin posuere sapien. Nulla varius metus id nibh volutpat bibendum. Ut non molestie lectus. Aenean at elementum tellus, feugiat condimentum orci. Nulla quis lacinia mauris, eu ullamcorper nisi. Suspendisse porttitor sollicitudin vehicula. Vivamus consectetur imperdiet elit, sit amet faucibus diam bibendum at. Donec aliquet, ligula malesuada blandit ultrices, nibh orci laoreet felis, quis vulputate est enim at odio. Quisque tellus arcu, dapibus nec faucibus a, pharetra non dui. Quisque lacus lectus, pulvinar et aliquet non, congue sit amet ipsum. | a  at  at  at  et  et  et  et  id  id  in  in  mi  ut  ut  ut  est  nec  non  non  non  non  non  sed  sit  sit  sit  sit  sit  cras  eget  nibh  nibh  nisi  nisi  nunc  dolor  dolor  magna  massa  semper  commodo | a  at  at  at  et  et  et  et  id  id  in  in  mi  ut  ut  ut  est  nec  non  non  non  non  non  sed  sit  sit  sit  sit  sit  cras  eget  nibh  nibh  nisi  nisi  nunc  dolor  dolor  magna  massa  semper  commodo |

Исходный код программы:

**import** re

inp = **input**()

words = [\_.lower() **for** \_ **in** re.split("\W", inp) **if** \_ != "" **and** **len**(**set**(re.findall("[аеёиоуыэюяaeiouy]",\_.lower())))==1]

words.sort(key=**lambda** x:(**len**(x),x))

**print**("\n".join(words))