

Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3

Регулярные выражения

Вариант №5_3_3_5_3

Выполнил

Макогон Ярослав Вадимович

Номер группы: P3118

Проверила

Малышева Т. А.

Содержание

Задание.....	3
Основные этапы вычислений.....	5
Заключение.....	11
Список использованных источников	12

Задание

Обязательное задание:

1. Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.
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	=

Рисунок 1. Таблица вариантов смайликов

Вариант смайлика по номеру ИСУ (466595):

466595 % 6 = 5 => глаза: [

466595 % 4 = 3 => нос: <{

466595 % 8 = 3 => рот: |

Смайлик: [<{|

Дополнительное задание:

1. Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.
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. Пример тестов приведён в таблице.

3. Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

5	С помощью регулярного выражения найти в тексте все слова, в которых две гласные стоят подряд, а после этого слова идёт слово, в котором не больше 3 согласных.	
	Пример:	
	Ввод	Вывод
	Кривошеее существо гуляет по парку	гуляет

Рисунок 2. Дополнительное задание 1

1. Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.
2. Для своей программы придумайте минимум 5 тестов. Все 5 тестов необходимо показать при защите.
3. Протестируйте свою программу на этих тестах.
4. Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

3	Студент Вася очень любит курс «Компьютерная безопасность». Однажды Васе задали домашнее задание зашифровать данные, переданные в сообщении. Недолго думая, Вася решил заменить все целые числа на функцию от этого числа. Функцию он придумал не сложную $4x^2 - 7$, где x – исходное число. Помогите Васе с его домашним заданием.
---	--



	Предусмотреть ситуацию, когда в тексте будут не только целые числа, но и любые символы.	
	Пример:	
	Ввод	Вывод
	20 + 22 = 42	1593 + 1929 = 7049

Рисунок 3. Дополнительное задание 2

Основные этапы вычислений

Обязательное задание:

Написана программа на Python, выполняющая поставленное задание:

```
import re

def count_emoji(emoji, text):
    return len(re.findall(r"\\[<\\{|\\|", text))

input_string = str(input())
emoji = "[<{|"

number_of_emoji = count_emoji(emoji, input_string)
print("Количество смайликов ", emoji, " : ", number_of_emoji)
```

Тесты для проверки корректности работы программы:

1. $[[\langle \{ \langle \{ \langle \{ \} \rangle \} \langle \{ \} \rangle -> \langle \{ \langle \{ \} = \langle \{ \}]$ - ответ 5

Ответ программы:

[[[<{<{|<{<{|}<{|-><{|<{|=<{|
Количество смайликов <{| : 5

Рисунок 4. Ответ программы на тест 1 задания 1

2. $[\{ \}] > [< [\{ \}] >] [< \{ \} >] < \{ \} > [\{ \}] [< \{ \} > |] > [< \{ \} >] [< | >] [< \{ \} >] [< \{ \} > |] [< \{ \} >] [< \{ \} >]$
 $[< \{ \} >] [< \{ \} >]$ - ответ б

Ответ программы:

```
[{]}>[<(<{|}>)[<{}>>]<{}>[{]}<{|}>|]>[<{|}>][<{|}>[<|>][<{|}>][<{}>]<|)][<{|}>]>[<{}>][<{|}>][<{|}>]
```

Количество смайликов [<{|] : 6

Рисунок 5. Ответ программы на тест 2 задания 1

3. `[<{|*[]}<{|<{|<{|["|as"<{|[]45-}<{|[]<{|-av}[b {}]}<{|<{|;a /<{|a 7 "<{|<{|}\as'q%;4r -`
ответ 8

Ответ программы:

```
[<{|*[]}><{|<{|<{|["|as"[<{|[]45-]<{|[] --av][b {}]><{| ;a /<{|a 7 ' '
[<{| ]\as' ]q%;4r
Количество смайликов [<{| : 8
```

Рисунок 6. Ответ программы на тест 3 задания 1

Дополнительное задание:

Написана программа на Python выполняющая поставленную задачу в дополнительном задании 1:

```
import re

def find_combinations(input):

    pattern=r"\S*[ааиеёоуыэюяaeiou][ааиеёоуыэюяaeiou]\S*\s(?:?:[ааиеёоуыэюяaeiou]
    *[^ааиеёоуыэюяaeiou\s][ааиеёоуыэюяaeiou]*[^ааиеёоуыэюяaeiou\s][ааиеёоуыэюяaeiou]*
    *[^ааиеёоуыэюяaeiou\s][ааиеёоуыэюяaeiou]*\b)|(?:[ааиеёоуыэюяaeiou]*[^ааиеёоуыэюяaeiou]
    [ааиеёоуыэюяaeiou]*[^ааиеёоуыэюяaeiou\s][ааиеёоуыэюяaeiou]*\b)|(?:[ааиеёоуыэюяaeiou]*
    [ааиеёоуыэюяaeiou]*[^ааиеёоуыэюяaeiou\s][ааиеёоуыэюяaeiou]*\b)|(?:[ааиеёоуыэюяaeiou]
    +\b))"
    words = input.split()

    result = []
    index_of_last_match = -1
    delta = 0
    for i in range(0, len(words)):
        to_re = ' '.join(words)
        found_combination = re.search(pattern, to_re, flags= re.I)

        if found_combination:
            if index_of_last_match != found_combination.start() + delta:
                index_of_last_match = found_combination.start() + delta
                result.append(found_combination[0])

            delta += len(words[0]) + 1
            words.pop(0)

    return result

input_string = str(input())
combinations = find_combinations(input_string)

result = []
for combination in combinations:
    word = combination.split()[0]
    result_word = re.sub(r"^[^а-яё]", '', word, flags=re.I)
    result.append(result_word)
if len(result) != 0:
    print("Ответ: ", *result)
else:
    print("Ничего не найдено")
```

Тесты для проверки корректности работы программы:

1. Яркие лучи солнца заливают садовые аллеи – ответ: Яркие заливают садовые

```
Яркие лучи солнца заливают садовые аллеи
Ответ: Яркие заливают садовые
```

Рисунок 9. Ответ программы на тест 1 задания 2

2. basdb bajsd babsdjb bbfoow bow pd jso j23pn nwin ppfwepaos – ответ: bbfoow

```
basdb bajsd babsdjb bbfoow bow pd jso j23pn nwin ppfwepaos
Ответ: bbfoow
```

Рисунок 10. Ответ программы на тест 2 задания 2

3. aa bbbaa bb aoa qp – ответ: aa bbbaa aoa

```
aa bbbaa bb aoa qp
Ответ: aa bbbaa aoa
```

Рисунок 11. Ответ программы на тест 3 задания 2

4. Тема: «Современные языки программирования. Python. Основы регулярных выражений.» - ответ: Современные

```
Тема: «Современные языки программирования. Python. Основы регулярных выражений.
»
Ответ: Современные
```

Рисунок 12. Ответ программы на тест 4 задания 2

5. В столовой, громадно-высокой, как и все комнаты в доме, ожидали выхода князя домашние и официанты, стоявшие за каждым стулом; дворецкий, с салфеткой на руке, оглядывал сервировку, мигая лакеям и постоянно перебегая беспокойным взглядом от стенных часов к двери, из которой должен был появиться князь. – ответ: домашние стоявшие мигая лакеям

```
В столовой, громадно-высокой, как и все комнаты в доме, ожидали выхода князя до
машние и официанты, стоявшие за каждым стулом; дворецкий, с салфеткой на руке,
оглядывал сервировку, мигая лакеям и постоянно перебегая беспокойным взглядом о
т стенных часов к двери, из которой должен был появиться князь.
Ответ: домашние стоявшие мигая лакеям
```

Рисунок 13. Ответ программы на тест 5 задания 2

Написана программа на Python выполняющая поставленную задачу в дополнительном задании 2:

```
import re

def encrypt_string(input):
    numbers = re.findall(r"\-?\d+", input)
    splited_string = re.split(r"\-?\d+", input)

    result = splited_string[0]
    for i in range(0, len(numbers)):
        change_to = str(int(numbers[i])**2 * 4 - 7)
        result += change_to + splited_string[i + 1]
    return result

input_string = str(input())
result = encrypt_string(input_string)
print("Ответ: ", result)
```

1. 12 + 12ans - 124 oqwe = 0120 – **ответ: 569 + 569ans - 61497 oqwe = 57593**

```
12 + 12ans - 124 oqwe = 0120
Ответ: 569 + 569ans - 61497 oqwe = 57593
```

Рисунок 14. Ответ программы на тест 1 задания 3

2. dnio124nkdsf4pn5i67ni4o3-0sd32ml[sd234owe – **ответ:**
dnio61497nkdsf57pn93i17949ni57o29--7sd4089ml[sd219017owe

```
dnio124nkdsf4pn5i67ni4o3-0sd32ml[sd234owe
Ответ: dnio61497nkdsf57pn93i17949ni57o29--7sd4089ml[sd219017owe
```

Рисунок 15. Ответ программы на тест 2 задания 3

3. as 2 gf3 2asd 1 3df 4 9 1dda – **ответ: as 9 gf29 9asd -3 29df 57 317 -3dda**

```
as 2 gf3 2asd 1 3df 4 9 1dda
Ответ: as 9 gf29 9asd -3 29df 57 317 -3dda
```

Рисунок 16. Ответ программы на тест 3 задания 3

4. iop12dsaop2e 1=1-2 142 s49mksda – **ответ: iop569dsaop9e -3=-39 80649**
s9597mksda

```
iop12dsaop2e 1=1-2 142 s49mksda
Ответ: iop569dsaop9e -3=-39 80649 s9597mksda
```

Рисунок 17. Ответ программы на тест 4 задания 3

5. q1xcv23sdf456eqw7qew89weq0 jo0123 pqw[=32]64 12 – **ответ: q-3xcv2109sdf831737eqw189qew31677weq-7 jo60509 pqw[=4089]16377 569**

```
q1xcv23sdf456eqw7qew89weq0 jo0123 pqw[=32]64 12
Ответ: q-3xcv2109sdf831737eqw189qew31677weq-7 jo60509 pqw[=4089]16377 569
```

Рисунок 18. Ответ программы на тест 5 задания 3

Заключение

- Были изучены принципы работы регулярных выражений
- Прделана практика по работе с регулярными выражениями на языке программирования Python
- Выучены основные обозначения и механики регулярных выражений

Список использованных источников

1. Статья “Регулярные выражения в Python от простого к сложному.” – URL: https://habr.com/ru/articles/349860/#Primer_ispolzovaniya_vseh_osnovnyh_funktsiy
2. Статья “Python RegEx: практическое применение регулярок” – URL: <https://tproger.ru/translations/regular-expression-python>