

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет Программной инженерии и компьютерной техники

Лабораторная работа №3
“Регулярные выражения”

Вариант №283566

Группа: Р3131
Выполнил: Друян Э.А.

Санкт-Петербург
2021г.

Текст заданий

Задание на 60 баллов (Смайлики)

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.
3. Программа должна считать количество смайликов определенного 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 % 5	Глаза	Номер в ИСУ % 4	Нос	Номер в ИСУ % 7	Рот
0	:	0	-	0	(
1	;	1	<	1)
2	X	2	-{	2	O
3	8	3	<{	3	
4				4	\
				5	/
				6	P

Пример смайлика: 8<{P

4. * нарисовав смайлик по вашему варианту при помощи средств языка программирования Python, можно заработать дополнительные баллы.

Выполнение задания на 60 баллов

```
import re

my_isu = 283566
eyes = [":", ";", "X", "8", "="]
noses = ["-", "<", "-{", "<{"];
mouths = ["(", ")", "0", "|", "\\ ", "/", "P"]

eye_dv = 5      # eye division remainder
nose_dv = 4     # nose division remainder
mouth_dv = 7    # mouth division remainder

def make_smile(isu_num):
    s = ""
    s += eyes[isu_num % eye_dv]
    s += noses[isu_num % nose_dv]
    s += mouths[isu_num % mouth_dv]
    return s

tests = [
    ';<{| ;<{\ \ :-{P :<{\ \ ', # 4
    ';-{P ;-{P =<{ ;-{P ',      # 2
    '=-{| 8-( 8--{| 8-{| ',      # 3
    ';<)<)<)<)<)<)<)<)<)<)<)', # 1
    '\_(o_0)_/ 8o'               # 0
]

def solve():
    my_re = r'[:;X8=]{1}[-<]{1}{0,1}[( )0|\ \ /P]{1}'
    for i in range(len(tests)):
        print(f"Test#{i + 1}:\t" + tests[i])
        d = {}
        for j in re.findall(my_re, tests[i]):
            if (j in d):
                d[j] += 1
            else:
                d[j] = 1
        print(f"N = {len(d)}:\t", end="")
        for j in d:
            print(j, end=" ")
        print()

def main():
    print("Smile of my ISU number:")
    print(make_smile(my_isu))

    solve()

if __name__ == "__main__":
    main()
```

Доп. задание №1 (+18 баллов)

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.

Пример тестов приведён в таблице ($283566 \% 6 = 0$).

Номер в ИСУ % 6	Задание								
0	<p>Хайку – жанр традиционной японской лирической поэзии века, известный с XIV века.</p> <p>Оригинальное японское хайку состоит из 17 слогов, составляющих один столбец иероглифов. Особыми разделительными словами – кирэдзи – текст хайку делится на части из 5, 7 и снова 5 слогов. При переводе хайку на западные языки традиционно вместо разделительного слова использую разрыв строки и, таким образом, хайку записываются как трёхстишия.</p> <p>Перед вами трёхстишия, которые претендуют на то, чтобы быть хайку. В качестве разделителя строк используются символы «/». Если разделители делят текст на строки, в которых 5/7/5 слогов, то выведите «Хайку!». Если число строк не равно 3, то выведите строку «Не хайку. Должно быть 3 строки.». Иначе выведите строку вида «Не хайку.»</p> <p>Для простоты будем считать, что слогов ровно столько же, сколько гласных, не задумываясь о тонкостях.</p> <p>Пример:</p> <table><tr><th>Ввод</th><th>Вывод</th></tr><tr><td>Вечер за окном. / Еще один день прожит. / Жизнь скоротечна...</td><td>Хайку!</td></tr><tr><td>Просто текст</td><td>Не хайку. Должно быть 3 строки.</td></tr><tr><td>Как вишня расцвела! / Она с коня согнала / И князя-гордеца.</td><td>Не хайку.</td></tr></table>	Ввод	Вывод	Вечер за окном. / Еще один день прожит. / Жизнь скоротечна...	Хайку!	Просто текст	Не хайку. Должно быть 3 строки.	Как вишня расцвела! / Она с коня согнала / И князя-гордеца.	Не хайку.
Ввод	Вывод								
Вечер за окном. / Еще один день прожит. / Жизнь скоротечна...	Хайку!								
Просто текст	Не хайку. Должно быть 3 строки.								
Как вишня расцвела! / Она с коня согнала / И князя-гордеца.	Не хайку.								

Выполнение доп. задания №1

```
import re

def main():
    line = input()
    arr = line.split(r'/')
    if len(arr) != 3:
        print("Не хайку. Должно быть 3 строки.")
        return

    pattern = "[ауоыиэяюеАУОЫИЭЯЮЕ]"
    nums = [5, 7, 5]
    for i in range(len(arr)):
        n = len(re.findall(pattern, arr[i]))
        if n != nums[i]:
            print("Не хайку.")
            return
    print("Хайку!")

if __name__ == "__main__":
    main()

# Вечер за окном. / Еще один день прожит. / Жизнь скоротечна...
# Просто текст
# Как вишня расцвела! / Она с коня согнала / И князя-гордеца.
# Тест#1
# ауоыиэяюеАУОЫИЭЯЮЕ / ауоыиэяюеАУОЫИЭЯЮЕ / ауоыиэяюеАУОЫИЭЯЮЕ
# Тест#2
# ауЭЯЮ / еАУОЫИЕ / ИЭЯЮЕ
# Тест#3
# ауЭЯЮ / еАУОЫИЕ
# Тест#4
# / /
# Тест#5
# Лень тест составлять / Это ведь так скучно, да? / Однозначно да!
```

Доп. задание No2 (+22 баллов)

1. Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.
2. Для своей программы придумайте минимум 5 тестов.
3. Протестируйте свою программу на этих тестах.
 $283566 \% 4 = 2$.

2

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 $3x^2 + 5$, где x – исходное число. Помогите Васе с его домашним заданием.

Пример:

Ввод	Вывод
$20 + 22 = 42$	$1205 + 1457 = 5297$

Выполнение доп. задания №2 (+22 балла)

```
import re

def main():
    line = input()
    pattern = r'[0-9]+'
    arr = re.findall(pattern, line)

    res_line = ""
    i = 0
    while i < len(line):
        if line[i] >= "0" and line[i] <= "9":
            j = i + 1
            while j < len(line) and line[j] >= "0" and line[j] <= "9":
                j += 1
            x = int(line[i:j])
            x = 3 * x * x + 5
            res_line += str(x)
            i = j
            continue
        else:
            res_line += line[i]
        i += 1
    print(res_line)

if __name__ == "__main__":
    main()

# Тест#1
# 21 + 22 = 42
# 1328 + 1457 = 5297
# Тест#2
# 123 4567
# 45392 62572472
# Тест#3
# 2 + 2 = 5
# 17 + 17 = 80
# Тест#4
# 2.3 + 8.7 = 0.0
# 17.32 + 197.152 = 5.5
# Тест#5
# Yes or no? True(1) or False(0)? Hmm...
# Yes or no? True(8) or False(5)? Hmm...
```

Консольные выводы программ

```
MacBook-Pro-ED:lab3 ed$ python lab3.py
Smile of my ISU number (283566):
;-{|
Test#1:      :<{| ;<{\ :-{P :<{\
N = 4: :<{| ;<{\ :-{P :<{\
Test#2:      ;-{P ;-{P =<C ;-{P
N = 2: ;-{P =<C
Test#3:      =-{| 8-( 8--{| 8-{|
N = 3: =-{| 8-( 8-{|
Test#4:      :<) :<) :<) :<)
N = 1: :<)
Test#5:      \_(o_0)_/ 8o
N = 0:
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
Вечер за окном. / Еще один день прожит. / Жизнь скоротечна...
Хайку!
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
Просто текст
Не хайку. Должно быть 3 строки.
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
Как вишня расцвела! / Она с коня согнала / И князя-гордеца.
Не хайку.
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
ауоыиэяюеАУОЫИЭЯЮЕ / ауоыиэяюеАУОЫИЭЯЮЕ / ауоыиэяюеАУОЫИЭЯЮЕ
Не хайку.
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
ауЭЯЮ / еАУОЫИЕ / ИЭЯЮЕ
Хайку!
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
ауЭЯЮ / еАУОЫИЕ
Не хайку. Должно быть 3 строки.
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
/ /
Не хайку.
MacBook-Pro-ED:lab3 ed$ python lab3_2.py
Лень тест составлять / Это ведь так скучно, да? / Однозначно да!
Хайку!
MacBook-Pro-ED:lab3 ed$ python lab3_3.py
21 + 22 = 42
1328 + 1457 = 5297
MacBook-Pro-ED:lab3 ed$ python lab3_3.py
123 4567
45392 62572472
MacBook-Pro-ED:lab3 ed$ python lab3_3.py
2 + 2 = 5
17 + 17 = 80
MacBook-Pro-ED:lab3 ed$ python lab3_3.py
2.3 + 8.7 = 0.0
17.32 + 197.152 = 5.5
MacBook-Pro-ED:lab3 ed$ python lab3_3.py
Yes or no? True(1) or False(0)? Hmm...
Yes or no? True(8) or False(5)? Hmm...
```


Задание при защите

5. Создать запрос для выделения текста, заключенного в кавычки. При решении задачи учесть, что текст может быть расположен на нескольких строках).

Выполнение задания при защите

```
import re

# ввод с учетом новой строки
n = int(input("Amount of lines and a text:\n"))
text = ""
for i in range(n):
    text += input() + " "
text = text[:-1] # deleting last "\n"

# использован метод re.split(pattern, string)
# 2-ой элемент полученного массива будет искомой строкой
arr = re.split(r'\"', text)
print("Output:\n" + arr[1])
```

Консольный вывод программы:

```
MacBook-Pro-ED:test_python ed$ python test.py
Amount of lines and a text:
2
Какой-то "очень
интересный текст!"
Output:
очень интересный текст!
```

Заключение

Ознакомился с базовым синтаксисом языка Python. Реализовал программные продукты на языке Python, используя регулярные выражения.