

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6
дисциплины «Программирование на Python»

Выполнил:

Лейс Алексей Вячеславович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики: кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Тема: Работа со строками в языке Python

Цель работы: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Индивидуальное задание:

Дано предложение. Определить, сколько в нем одинаковых соседних букв.

```
1.py X
E: > ИВТ-6-о-22-1 > Программирование на Python > 6 > 1.py > ...
1 #Дан текст. Сколько раз в нем встречается символ «
2 print("Введите текст")
3 string = ""
4 string = input()
5 count = 0
6 for i in string:
7     if(i == "+" or i == "***"):
8         count +=1
9
10 print(f"Количество + и *: {count}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation), 2016. Все права защищены.

PS E:\ИВТ-6-о-22-1\Программирование на Python\6> & 'C:\Users\Student\AppData\Local\Programs\Python\Python36\launcher' '55154' '--' 'e:\ИВТ-6-о-22-1\Программирование на Python\6\1.py'

Введите текст
ghbdtnпривет+***+fadsfasdf+***+a*

PS E:\ИВТ-6-о-22-1\Программирование на Python\6> e:; cd 'e:\ИВТ-6-о-22-1\Программирование на Python\6\1.py'

Введите текст
ghbdtnпривет+***+ghdjasf

Количество + и *: 6

Дано предложение. Заменить в нем все вхождения буквосочетания про на нет.

```
> ИВТ-6-о-22-1 > Программирование на Python > 6 > 2.py > ...
1 #2
2 #Дано предложение. Заменить в нем все вхождения буквосочетания про на нет.
3 string = "проехала пробежала прошла про-икроки - процессор".replace("про", "нет")
4 print(string)
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation), 2016. Все права защищены.

PS E:\ИВТ-6-о-22-1\Программирование на Python\6> & 'C:\Users\Student\AppData\Local\Programs\Python\Python36\launcher' '55217' '--' 'e:\ИВТ-6-о-22-1\Программирование на Python\6\2.py'

проехала нетбежала нетшла нет-икроки - нетцессор

PS E:\ИВТ-6-о-22-1\Программирование на Python\6> █

Дано ошибочно написанное слово роцессорп. Путем перемещения его букв получить слово процессор.

```
E: > ИВТ-6-о-22-1 > Программирование на Python > 6 > 4.py > ...
1  #3
2  #Дано ошибочно написанное слово роцессорп. Путем перемещения его
3  #процессор.
4  word = "роцессорп"
5  print("Слово: ",word)
6  list = []
7  for i in word:
8      list.append(i)
9  print("По буквам: " ,list)
10 try_word=["g"]
11 b = int(len(list)-1)
12 a =list[b]
13 try_word[0] = a
14 count = 0
15 for l in list:
16     count+=1
17     if count != b+1:
18         try_word.append(l)
19 r = " "
20 print("По буквам: " ,try_word)
21 for i in try_word:
22     r +=i
23 print("Слово: " ,r)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation), 2016. Все права защищены.

PS E:\ИВТ-6-о-22-1\Программирование на Python\6> & 'C:\Users\Student\AppData\Local\Microsoft\Windows\Apps\Windows Defender Security Center\Windows Defender Security Center.exe' 'c:\Users\Student\vscode\extensions\ms-python.python-2021.12.1559732659\bin\python.exe' 'e:\ИВТ-6-о-22-1\Программирование на Python\6\4.py'

Слово: роцессорп
По буквам: ['p', 'o', 'c', 'e', 's', 's', 'o', 'r', 'p', 'n']
По буквам: ['n', 'p', 'o', 'c', 'e', 's', 's', 'o', 'r']
Слово: процессор
PS E:\ИВТ-6-о-22-1\Программирование на Python\6>

Повышенной сложности:

Дано предложение. Найти длину его самого короткого слова.

```
1  # 19. Дано предложение. Найти длину его самого короткого слова.
2  str = "Дано предложение, найти длину его самого короткого слова." #его
3  arr_str = []
4  for i in str:
5      arr_str.append(i)
6
7  count = int(0)
8  min_index = -1
9  min = 99
10 for i in range(0,len(arr_str) - 1):
11     if arr_str[i] != " ":
12         count += 1
13     if arr_str[i] == " ":
14         if min > count:
15             min = count
16             min_index = i - 1
17         count = 0
18 word = ""
19 for i in range (0, min):
20     word += arr_str[min_index - min+1 + i ]
21 print(f"Полученное словов: {word}")
22
```

input

Полученное словов: его

Ответы на вопросы:

1. Что такое строки в языке Python?

- Строки в Python представляют собой последовательность символов и используются для представления текстовой информации.

2. Какие существуют способы задания строковых литералов в языке Python?

- Строковые литералы могут быть заданы с использованием одинарных (' '), двойных (" "), или тройных (" " " или "" "" """) кавычек.

3. Какие операции и функции существуют для строк?

- Существует множество операций и методов для строк, таких как конкатенация (+), умножение (*), функции len(), методы upper(), lower(), и другие.

4. Как осуществляется индексирование строк?

- Индексирование строк начинается с 0, и символ в строке может быть получен по его индексу, например, `my_string[0]` вернет первый символ строки.

5. Как осуществляется работа со срезами для строк?

- Срезы в Python создаются с использованием оператора `:`. Например, `my_string[1:4]` возвращает подстроку с 1-го по 3-й символ строки.

6. Почему строки Python относятся к неизменяемому типу данных?

- Строки являются неизменяемыми, потому что после создания строки нельзя изменить ее содержимое. Можно создать новую строку с необходимыми изменениями.

7. Как проверить то, что каждое слово в строке начинается с заглавной буквы?

- Можно воспользоваться методом `istitle()`, который возвращает `True`, если каждое слово начинается с заглавной буквы.

8. Как проверить строку на вхождение в неё другой строки?

- Можно воспользоваться оператором `in`. Например, `substring in my_string` вернет `True`, если подстрока присутствует в строке.

9. Как найти индекс первого вхождения подстроки в строку?

- Метод `find()` возвращает индекс первого вхождения подстроки, или `-1`, если подстрока не найдена.

10. Как подсчитать количество символов в строке?

- Функция `len(my_string)` возвращает количество символов в строке.

11. Как подсчитать то, сколько раз определённый символ встречается в строке?

- Метод `count()` позволяет подсчитать количество вхождений определённого символа в строке.

12. Что такое f-строки и как ими пользоваться?

- F-строки - это способ форматирования строк, позволяющий вставлять значения переменных в строку прямо в месте её объявления. Используется префикс `f` перед строкой.

13. Как найти подстроку в заданной части строки?

- Можно воспользоваться методом `find()` с указанием начального и конечного индексов.

14. Как вставить содержимое переменной в строку, воспользовавшись методом `format()`?

- С использованием метода `format()`, переменные могут быть вставлены в строку. Например, `"Привет, {}!".format(name)`.

15. Как узнать о том, что в строке содержатся только цифры?

- Метод `isdigit()` возвращает `True`, если все символы строки являются цифрами.

16. Как разделить строку по заданному символу?

- Метод `split()` разделяет строку на подстроки, используя заданный разделитель.

17. Как проверить строку на то, что она составлена только из строчных букв?

- Метод `islower()` возвращает `True`, если все буквы в строке строчные.

18. Как проверить то, что строка начинается со строчной буквы?

- Метод `islower()` также может быть использован для проверки, что первая буква строки строчная.

19. Можно ли в Python прибавить целое число к строке?

- Нет, строки и числа не могут быть сложены напрямую. Необходимо преобразовать число в строку или наоборот.

20. Как «перевернуть» строку?

- С использованием среза `[::-1]`, например, `reversed_string = my_string[::-1]`.

21. Как объединить список строк в одну строку, элементы которой разделены дефисами?

- Метод `join()` может быть использован: `"-".join(my_list)`.

22. Как привести всю строку к верхнему или нижнему регистру?

- Методы `upper()` и `lower()` соответственно изменяют строку на верхний и нижний регистр.

23. Как преобразовать первый и последний символы строки к верхнему регистру?

- Можно воспользоваться методами `capitalize()` для первого символа и `title()` для каждого слова.

24. Используя методы `'capitalize()'` для первого символа и `'upper()'` для последнего:

```
```python
string = string[0].capitalize() + string[1:-1] + string[-1].upper()
```
```

25. С помощью метода `'islower()'`:

```
```python
is_lower = my_string.islower()
```
```

26. Метод `'splitlines()'` используется для разделения строки на список строк по символу новой строки.

27. С использованием метода `'replace()'`:

```
```python
new_string = original_string.replace(substring_to_replace, replacement_string)
```
```

28. Для проверки начала или конца строки используются методы `'startswith()'` и `'endswith()'` соответственно.

29. Чтобы проверить, что строка состоит только из пробелов, используйте метод `'isspace()'`:

```
```python
contains_only_spaces = my_string.isspace()
```
```

30. Строка будет повторена три раза.

31. Используя метод `'title()'`:

```
```python
```

```
new_string = original_string.title()
```

```
...
```

32. Метод `'partition()'` разделяет строку на три части по первому вхождению указанной подстроки и возвращает кортеж.

33. Метод `'rfind()'` используется для поиска последнего вхождения подстроки в строке и возвращает индекс этого вхождения.

Вывод: на основе выполненной работы приобрели навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.