МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Кафедра инфокоммуникаций

Отчет

по лабораторной работе №2

«Работа со строками в языке Python»

по дисциплине:

«Введение в системы искусственного интеллекта»

Вариант 7

Выполнил: студент группы ИВТ-б-о-18-1 (2)
Криворучко Ярослав Евгеньевич
_____(подпись)
Проверил:
Воронкин Роман Александрович
_____(подпись)

Ставрополь, 2022 г.

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

Задание №1

7. Дано предложение. Напечатать все его буквы и.

```
Ввод [1]: string = str(input("Введите предложение: "))
for i in range(len(string)):
    if(string[i] == 'и'):
        print(string[i])
```

```
Введите: Ириска и кино
и
и
и
```

Рисунок 1 – Листинг программы

Задание №2

7. Дано предложение. Определить количество букв н, предшествующих первой запятой предложения. Рассмотреть два случая:

- известно, что запятые в предложении есть;
- запятых в предложении может не быть.

```
Ввод [1]: string = str(input("Введите предложение: "))
start = string.find(',')

if(string.count(',')!= 0):
   print(string.count('н',start))
else:
    print("Запяых в предложении нет!")
```

```
Введите предложение: Номер нулевой, нужно найти номер
4
```

Рисунок 2 – Листинг программы

Задание №3

7. Дано слово. Удалить из него все повторяющиеся буквы, оставив их первые вхождения, т. е. в слове должны остаться только различные буквы.

```
Ввод [8]: word = input("Введите слово: ")
print("".join(sorted(set(word), key=word.index)))
```

```
Введите слово: Аарпп
Аарп
```

Рисунок 3 – Листинг программы

Задание повышенной сложности

7. Даны два слова. Напечатать только те буквы слов, которые есть лишь в одном из них (в том числе повторяющиеся). Например, если заданные слова процессор и информация, то ответом должно быть: п е с с и ф м а я.

Ввод [19]:
```
word1 = str(input("Введите первое слово: "))
word2 = str(input("Введите второе слово: "))

n=0
x = word1 + word2
new_x = ''

for i in range(0,len(word1)):
   for j in range(0,len(word2)):
      if(word1[i]==word2[j]):
         n+=1
   if(n==0):
      new_x += word1[i] + ' '
   n=0

for i in range(0,len(word2)):
   for j in range(0,len(word1)):
      if(word2[i]==word1[j]):
         n+=1
   if(n==0):
      new_x += word2[i] + ' '
   n=0

print(new_x)
```

```
Введите первое слово: процессор
Введите второе слово: информация
п е с с и н ф м а и я
```

Рисунок 4 – Листинг программы

Файл 2.3.ipynb с решением задач находится на **Github**: https://github.com/YaroStavr/LR2_Artificial-Intelligence.git

Ответы на вопросы:

1. Что такое строки в языке Python?

Строки в Python – упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

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

Строки в апострофах и в кавычках: c = ‘test”s’, c = “test’s”

Строки в апострофах и в кавычках - одно и то же. Причина наличия двух вариантов в том, чтобы позволить вставлять в литералы строк символы кавычек или апострофов, не используя экранирование.

3. Какие операции и функции существуют для строк?

+ — оператор конкатенации строк. Он возвращает строку, состоящую из других строк.

* — оператор создает несколько копий строки. Если s это строка, а n целое число, любое из следующих выражений возвращает строку, состоящую из n объединенных копий s.

Python также предоставляет оператор принадлежности, который можно использоваться для манипуляций со строками. Оператор in возвращает True , если подстрока входит в строку, и False , если нет.

Функция chr() Преобразует целое число в символ

Функция ord() Преобразует символ в целое число

Функция len() Возвращает длину строки

Функция str() Изменяет тип объекта на string

4. Как осуществляется индексирование строк?

Доступ к символам в строках основан на операции индексирования – после строки или имени переменной, ссылающейся на строку, в квадратных скобках указываются номера позиций необходимых символов.

Так же следует понимать, что этот самый доступ, основан на смещении, т.е. расстоянии символов от левого или правого края строки. Данное расстояние измеряется целыми числами и по сути определяет номер позиции символов в строке – их индекс. Подвох заключается в словосочетании "измеряется целыми числами", а это означает, что индекс может быть как положительным так и отрицательным: положительные индексы – это отсчет от левого края, а отрицательные – от правого. Причем отсчет символов от левого края начинается с 0, а с правого начинается с −1 (минус единицы).

5. Как осуществляется работа со срезами для строк?

В Python строки являются упорядоченными последовательностями символьных данных и могут быть проиндексированы. Доступ к отдельным символам в строке можно получить, указав имя строки, за которым следует число в квадратных скобках [] .

Индексация строк начинается с нуля: у первого символа индекс 0 , следующего 1 и так далее. Индекс последнего символа в python — ‘‘длина строки минус один’’.

```
>>> s = 'foobar'

>>> s[0]
'f'
>>> s[1]
'o'
```

6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять.

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

string.istitle() определяет, начинаются ли слова строки с заглавной буквы.

s.istitle() возвращает True когда s не пустая строка и первый алфавитный символ каждого слова в верхнем регистре, а все остальные буквенные символы в каждом слове строчные. Возвращает False , если нет:

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

string.count(<sub>[, <start>[, <end>]]) подсчитывает количество вхождений подстроки в строку.

s.count(<sub>) возвращает количество точных вхождений подстроки <sub> в s

9. Как найти индекс первого вхождения подстроки в строку?

s.find(<sub>) возвращает первый индекс в s который соответствует началу строки <sub>

10. Как подсчитать количество символов в строке?

len(s) возвращает количество символов в строке s

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

s.count(<sub>) возвращает количество точных вхождений подстроки <sub> в s

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

Эта функция официально названа литералом отформатированной строки, но обычно упоминается как f-строки (f-string).

Одной простой особенностью f-строк, которые вы можете начать использовать сразу, является интерполяция переменной. Вы можете указать имя переменной непосредственно в f-строковом литерале (f'string'), и python заменит имя соответствующим значением.

13. Как найти подстроку в заданной части строки?

string.find(<sub>[, <start>[, <end>]]) ищет в строке заданную подстроку.

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

'Hello, {}!'.format('Vasya')

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

string.isdigit() определяет, состоит ли строка из цифр (проверка на число).

16. Как разделить строку по заданному символу?

s.join(<iterable>) возвращает строку, которая является результатом конкатенации объекта <iterable> с разделителем s .

', '.join(['foo', 'bar', 'baz', 'qux'])

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

string.islower() определяет, являются ли буквенные символы строки строчными.

s.islower() возвращает True , если строка s не пустая, и все содержащиеся в нем буквенные символы строчные, а False если нет. Не алфавитные символы игнорируются.

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

'aPPLE'[0].islower() #=> True

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

При попытке выполнения подобной операции будет выдана ошибка TypeError.

20. Как «перевернуть» строку?

Для того чтобы «перевернуть» строку, её можно разбить, представив в виде списка символов, «перевернуть» список, и, объединив его элементы, сформировать новую строку.

''.join(reversed("hello world"))

#=> 'dlrow olleh'

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

'-'.join(['a','b','c'])

#=> 'a-b-c'

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

sentence = 'The Cat in the Hat'

sentence.upper() #=> 'THE CAT IN THE HAT'

sentence.lower() #=> 'the cat in the hat'

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

animal = 'fish'

animal[0].upper() + animal[1:-1] + animal[-1].upper()

#=> 'FisH'

24. Как проверить строку на то, что она составлена только из прописных букв?

isupper() возвращает True только в том случае, если вся строка состоит из прописных букв.

25. В какой ситуации вы воспользовались бы методом splitlines() ?

Метод splitlines() разделяет строки по символам разрыва строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?

Методом replace().

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?

Методы startswith() и endswith().

city = 'New York'

city.startswith('New') #=> True

city.endswith('N') #=> False

28. Как узнать о том, что строка включает в себя только пробелы?

Метод isspace(), который возвращает True только в том случае, если строка состоит исключительно из пробелов

29. Что случится, если умножить некую строку на 3?

Будет создана новая строка, представляющая собой исходную строку, повторённую три раза.

30. Как привести к верхнему регистру первый символ каждого слова в строке?

s.title() возвращает копию, s в которой первая буква каждого слова преобразуется в верхний регистр, а остальные буквы — в нижний регистр:

31. Как пользоваться методом partition() ?

s.partition(<sep>) отделяет от s подстроку длиной от начала до первого вхождения <sep> .

Возвращаемое значение представляет собой кортеж из трех частей:

Часть s до <sep>

Разделитель <sep>

Часть s после <sep>

>>> 'foo.bar'.partition('.')

('foo', '.', 'bar')

>>> 'foo@@bar@@baz'.partition('@@')

('foo', '@@', 'bar@@baz')

32. В каких ситуациях пользуются методом rfind() ?

string.rfind(<sub>[, <start>[, <end>]]) ищет в строке заданную подстроку, начиная с конца.