

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.3

Дисциплина: «Основы кроссплатформенного программирования»

Тема: **«Работа со строками в языке Python3»**

Выполнил: студент 1 курса

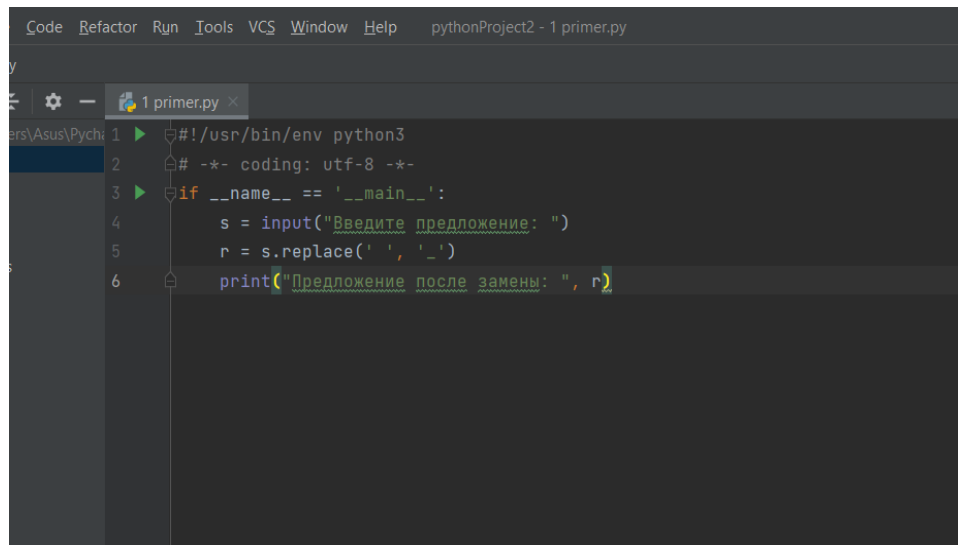
группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

Ставрополь 2022

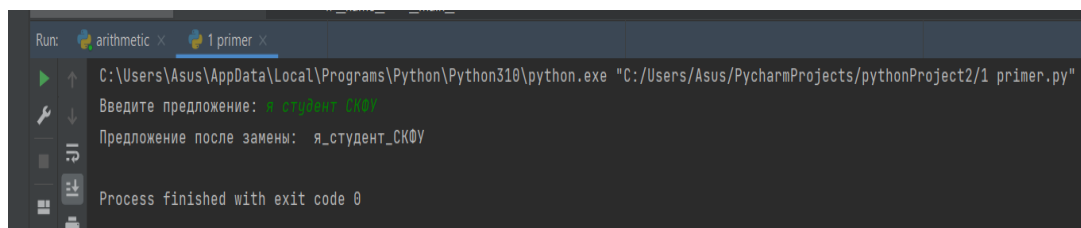
Ход работы:

Пример 1. Дано предложение. Все пробелы в нем заменить символом «_».



```
Code Refactor Run Tools VCS Window Help pythonProject2 - 1 primer.py
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 if __name__ == '__main__':
4     s = input("Введите предложение: ")
5     r = s.replace(' ', '_')
6     print("Предложение после замены: ", r)
```

Рисунок 1. Код программы



```
Run: arithmetic x 1 primer x
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe "C:/Users/Asus/PycharmProjects/pythonProject2/1 primer.py"
Введите предложение: я студент СКФУ
Предложение после замены: я_студент_СКФУ
Process finished with exit code 0
```

Рисунок 2. Результат выполнения программы

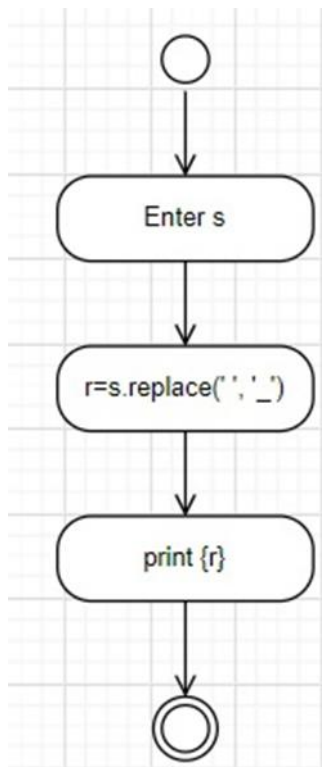


Рисунок 3. UML-диаграмма

Пример 2. Дано слово. Если его длина нечетная, то удалить среднюю букву, в противном случае – две средние буквы.

```

1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      word = input("Введите слово: ")
6
7      idx = len(word) // 2
8      if len(word) % 2 == 1:
9          # Длина слова нечетная.
10         r = word[:idx] + word[idx+1:]
11     else:
12         # Длина слова четная.
13         r = word[:idx-1] + word[idx+1:]
14     print(r)
  
```

Рисунок 4. Код программы

```

Run: C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe "C:/Users/Asus/PycharmProjects/pythonProject2/2
Введите слово: 06paz
06z
Process finished with exit code 0
  
```

Рисунок 5. Результат выполнения программы

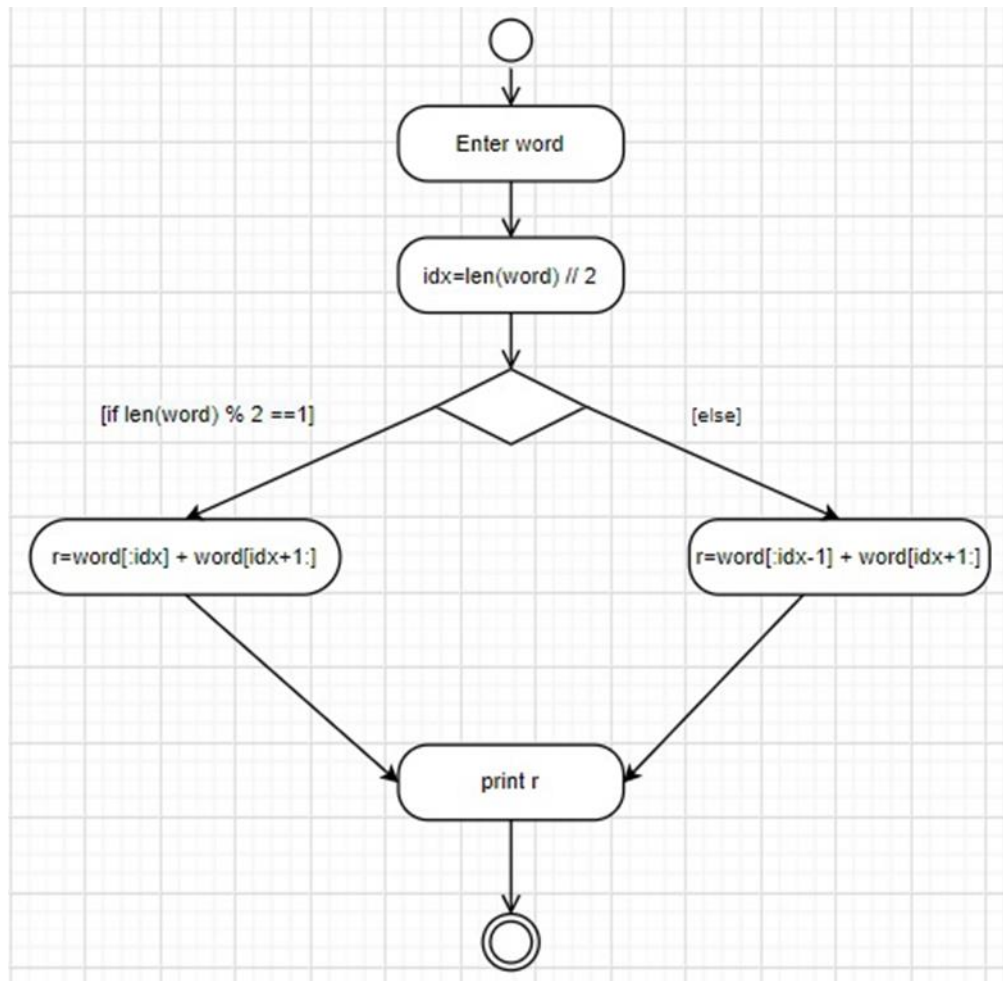


Рисунок 6. UML-диаграмма

Индивидуальные задания:

Задание 1. Дан текст. Сколько раз в нем встречается символ «+» и сколько раз символ «*»?

```

pythonProject2 - ind 1.py
pythonProject2 C:\Users\Asus\Pytho
ind 1.py
main.py
External Libraries
Scratches and Consoles

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == '__main__':
4      s = input("Введите строку: ")
5      print("Количество плюсов = ", s.count("+"), ", количество звездочек = ", s.count("*"))
  
```

Рисунок 7. Код программы

```
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe "C:/Users/Asus/PycharmProjects/pythonProject2/ind 1.py"
Введите строку: ++23*3+12**
Количество плюсов = 2 , количество звездочек = 2
Process finished with exit code 0
```

Рисунок 8. Результат выполнения программы

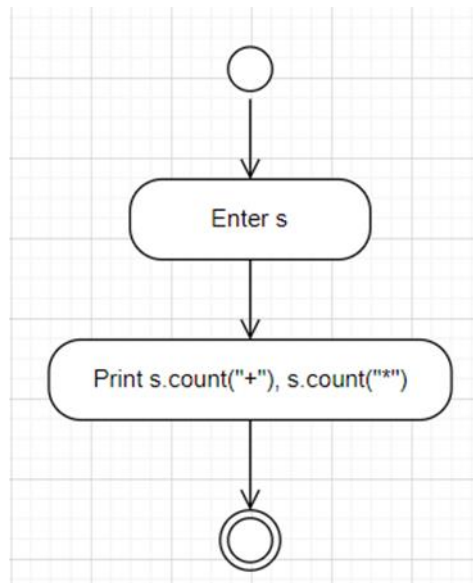


Рисунок 9. UML-диаграмма

Задание 2. Дано предложение. Заменить в нем все вхождения буквосочетания да на не.

```
pythonProject2 - ind 2.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == '__main__':
4      s = input("Введите предложение: ")
5      s = s.replace('да', 'не')
6      print(s)
```

Рисунок 10. Код программы

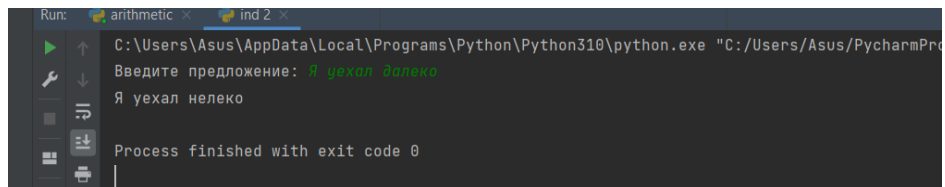


Рисунок 11. Результат выполнения программы

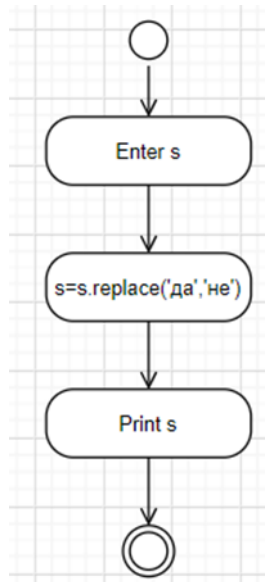


Рисунок 12. UML-диаграмма

Задание 3. Дано ошибочно написанное слово алигортм. Путем перемещения его букв получить слово алгоритм.

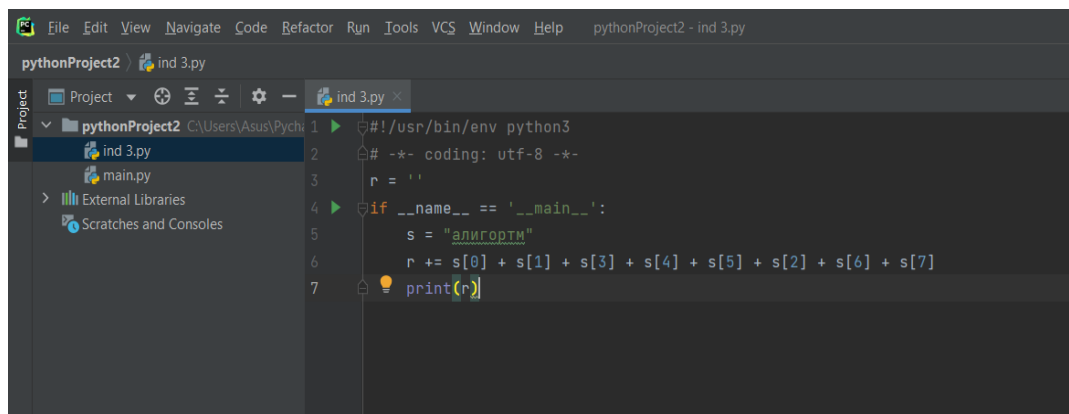


Рисунок 13. Код программы

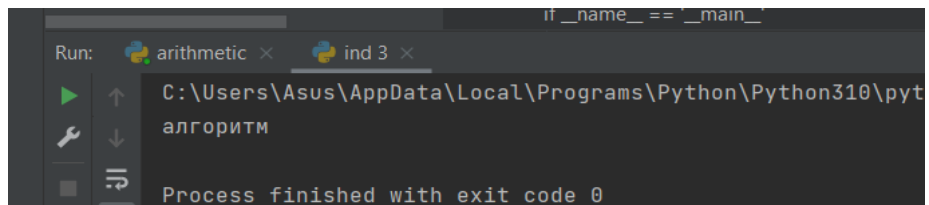


Рисунок 14. Результат выполнения программы

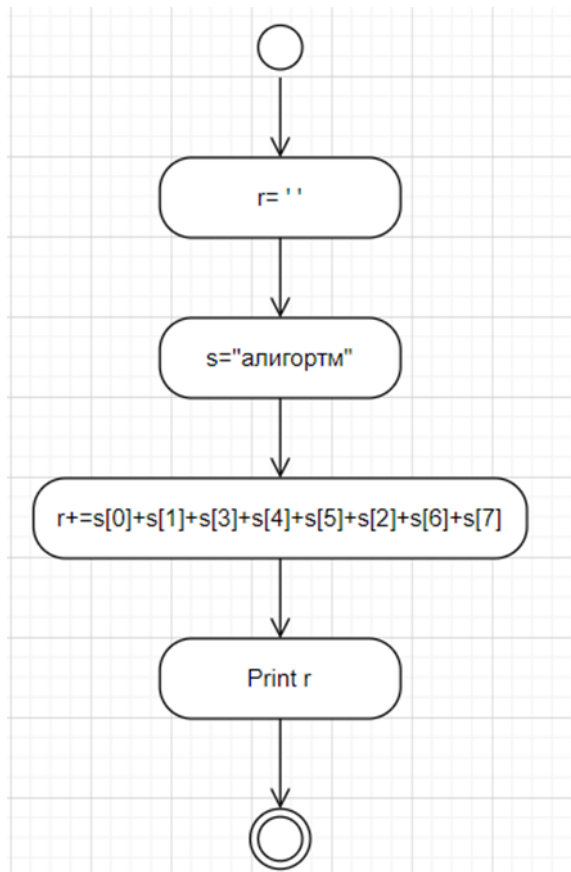
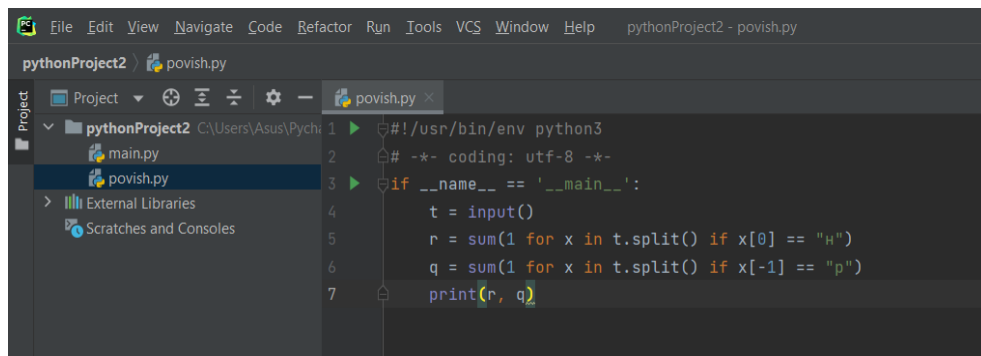


Рисунок 15. UML-диаграмма

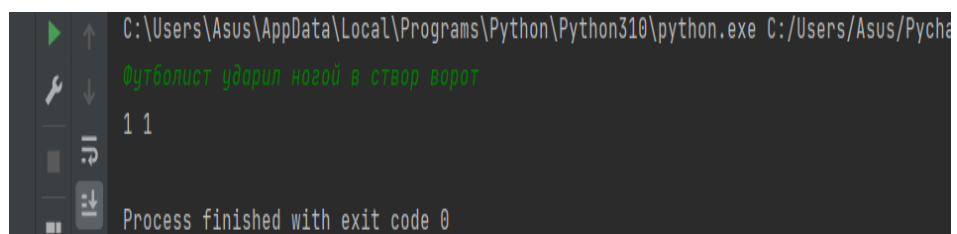
Задание повышенной сложности. Дано предложение. Определить количество слов:

- начинающихся с буквы н;
- оканчивающихся буквой р



```
pythonProject2 - povish.py
pythonProject2
Project
  pythonProject2
    main.py
    povish.py
  External Libraries
  Scratches and Consoles
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 if __name__ == '__main__':
4     t = input()
5     r = sum(1 for x in t.split() if x[0] == "h")
6     q = sum(1 for x in t.split() if x[-1] == "p")
7     print(r, q)
```

Рисунок 16. Код программы



```
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Asus/Pycha
Футболист ударил ногой в створ ворот
1 1
Process finished with exit code 0
```

Рисунок 17. Результат выполнения программы

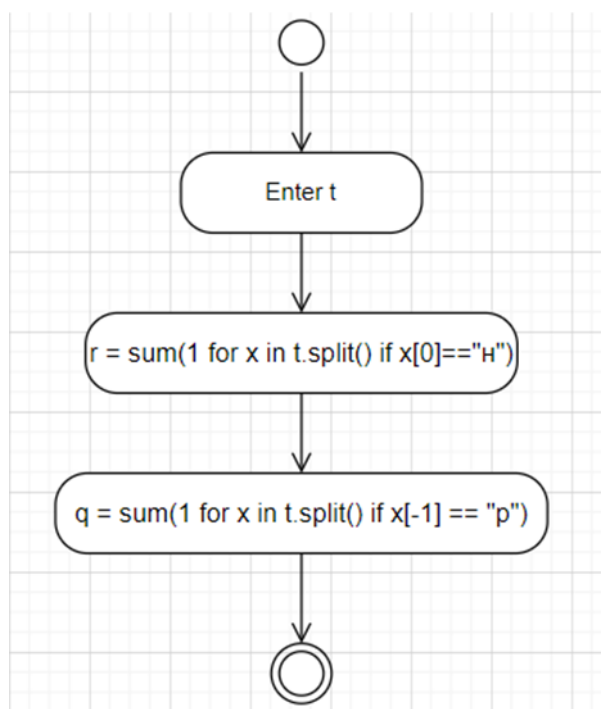


Рисунок 18. UML-диаграмма

Ответы на вопросы

Контр. вопросы и ответы на них:

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, экранированные последовательности, "сырые" строки, строки в тройных апострофах или кавычках

3. Какие операции и функции существуют для строк?

Сложение, дублирование, длина строки, извлечение среза и т. д.

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.

5. Как осуществляется работа со срезами для строк?

Есть три формы срезов. Самая простая форма среза: взятие одного символа строки, а именно, `S[i]` — это срез, состоящий из одного символа, который имеет номер `i`, при этом считая, что нумерация начинается с числа 0. То есть если `S = 'Hello'`, то `S[0]=='H'`, `S[1]=='e'`, `S[2]=='l'`, `S[3]=='l'`, `S[4]=='o'`. Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1.

Срез с двумя параметрами: `S[a:b]` возвращает подстроку из `b-a` символов, начиная с символа с индексом `a`, то есть до символа с индексом `b`, не включая его.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. Python дает возможность изменять (заменять и перезаписывать) строки.

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

`string.istitle()`

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

`string.find()`

9. Как найти индекс первого вхождения подстроки в строку?

`s.partition(<sep>)`

10. Как подсчитать количество символов в строке?

`len(s)`

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

`s.count(<sub>)`

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

Эти строки улучшают читаемость кода, а также работают быстрее чем другие способы форматирования. F-строки задаются с помощью литерала «f» перед кавычками. Пример: `print(f'Меня зовут {name} Мне {age} лет.')`

13. Как найти подстроку в заданной части строки?

`s.find(значение, начало, конец)`

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

`print('{} {}'.format(s))`

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

`s.isdigit()`

16. Как разделить строку по заданному символу?

`str.split()`

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

`s.isalpha()`

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

`s.istitle()`

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

Нет

20. Как «перевернуть» строку?

`s.reverse()`

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

`str.split('-')`

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

`s.upper()`

`s.lower`

23. Как преобразовать первый символ строки к верхнему регистру?

`s.capitalize()`

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

`s.isupper()`

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

`s.splitlines()` делит `s` на строки и возвращает их в списке. Любой из следующих символов или последовательностей символов считается границей строки.

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

`s.replace(old, new)`

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

`str.startswith()` и `str.endswith()`

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

`s.isspace()`

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

`Asd*3 = AsdAsdAsd`

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

`s.title()`

31. Как пользоваться методом `partition()`?

Метод `partition()` разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом `rfind()`?

`s.rfind(<sub>)` возвращает индекс последнего вхождения подстроки `<sub>` в `s`, который соответствует началу `<sub>`