

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2.3 по дисциплине основы
программной инженерии**

Выполнил:

Кожухов Филипп Денисович,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры
прикладной математики и
компьютерной
безопасности, Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

```
1 ▶  #!/usr/bin/env python3
2     # -*- coding: utf-8 -*-
3
4 ▶  if __name__ == '__main__':
5       s = input("Введите предложение: ")
6       r = s.replace(' ', '_')
7     print(f"Предложение после замены: {r}")
```

Рисунок 6.1 - Код примера 1

```
"C:\Users\CMDR Inferion\AppData\Local\Micro
Введите предложение: Пример работы
Предложение после замены: Пример_работы

Process finished with exit code 0
|
```

Рисунок 6.2 - Вывод программы

```
1 ▶  #!/usr/bin/env python3
2     # -*- coding: utf-8 -*-
3
4
5 ▶  if __name__ == '__main__':
6       word = input("Введите слово: ")
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)
15
```

Рисунок 6.3 - Код примера 2

```

"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsAp
Введите слово: Hello
Helo

Process finished with exit code 0
|

"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsA
Введите слово: World!
Wod!

Process finished with exit code 0
|

```

Рисунок 6.4 - Вывод программы

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import sys
5
6  ▶  if __name__ == '__main__':
7      s = input("Введите предложение: ")
8      n = int(input("Введите длину: "))
9      # Проверить требуемую длину.
10     if len(s) ≥ n:
11         print(
12             "Заданная длина должна быть больше длины предложения",
13             file=sys.stderr
14         )
15         exit(1)
16     # Разделить предложение на слова.
17     words = s.split(' ')
18     # Проверить количество слов в предложении.
19     if len(words) < 2:
20         print(
21             "Предложение должно содержать несколько слов",
22             file=sys.stderr
23         )
24         exit(1)
25     # Количество пробелов для добавления.
26     delta = n
27     for word in words:
28         delta -= len(word)
29     # Количество пробелов на каждое слово.
30     w, r = delta // (len(words) - 1), delta % (len(words) - 1)
31     # Сформировать список для хранения слов и пробелов.
32     lst = []
33     # Пронумеровать все слова в списке и перебрать их.
34     for i, word in enumerate(words):
35         lst.append(word)
36         # Если слово не является последним, добавить пробелы.
37         if i < len(words) - 1:
38             # Определить количество пробелов.
39             width = w
40             if r > 0:
41                 width += 1
42                 r -= 1
43             # Добавить заданное количество пробелов в список.
44             if width > 0:
45                 lst.append(' ' * width)
46
47     # Вывести новое предложение, объединив все элементы списка lst.
48     print(''.join(lst))
49

```

Рисунок 6.5 - Код примера 3

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsA
Введите предложение: Буря мглою небо кроет
Введите длину: 30
Буря мглою небо кроет

Process finished with exit code 0
|
```

Рисунок 6.6 - Вывод программы

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Вариант 5

Задание №1

```
1 ► #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4 ► if __name__ == '__main__':
5     word = input("Enter the word: ")
6     n = len(word)
7     print(f"The length of this word is: {n}")
8     word = n * '*' + word + n * '*'
9     print(word)
10  |
```

Рисунок 6.7 - Код задания 1

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsA
Enter the word: Python
The length of this word is: 6
*****Python*****

Process finished with exit code 0
|
```

Рисунок 6.8 - Вывод программы

Задание №2

```
1 ▶ #!/usr/bin/env python3  
2 # -*- coding: utf-8 -*-  
3  
4 ▶ if __name__ == '__main__':  
5     word1, word2 = input("Enter the two words: ").split()  
6     if word1 == word2:  
7         print("All of the characters from these words are the same!")  
8         exit(0)  
9     else:  
10         n = 0  
11         for i in range(len(word1)):  
12             if word1[i] in word2[i]:  
13                 n += 1  
14             else:  
15                 break  
16         print(f"There are {n} of the same first letters in these two words")  
17
```

Рисунок 6.9 - Код задания 2

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps  
Enter the two words: Space Shuttle  
There are 1 of the same first letters in these two words  
  
Process finished with exit code 0  
|
```

Рисунок 6.10 - Первые буквы разных слов совпадают

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps  
Enter the two words: coding coding  
All of the characters from these words are the same!  
  
Process finished with exit code 0  
|
```

Рисунок 6.11 - Одинаковые слова

Задание №3

```
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4 ▶   if __name__ == '__main__':
5       sentence = input("Enter the sentence: ")
6       sentence = sentence.replace('с', '') # Удаление английских букв "с"
7       sentence = sentence.replace('с', '') # Удаление русских букв "с"
8       print(sentence)
9
```

Рисунок 6.12 - Код задания 3

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\python3.8.exe" "C:/Users/CMDR In
Enter the sentence: На английском фраза мастера Йоды звучит так: May the Force be with you
На анлийком фраза матеря Йоды звучит так: May the Fore be with you

Process finished with exit code 0
|
```

Рисунок 6.13 - Вывод программы

ЗАДАНИЕ ПОВЫШЕННОЙ СЛОЖНОСТИ

Вариант 5

Даны два слова. Для каждой буквы первого слова (в том числе для повторяющихся в этом слове букв) определить, входит ли она во второе слово. Например, если заданные слова "информация" и "процессор", то для букв первого из них ответом должно быть: нет нет нет да да нет нет да нет нет.

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == '__main__':
5      word1, word2 = input("Enter the two words: ").split()
6      for ch in word1:
7          if ch in word2:
8              print("Yes", end=" ")
9          else:
10             print("No", end=" ")
11
```

Рисунок 6.14 - Код задания повышенной сложности

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\pyt
Enter the two words: информация процессор
Нет Нет Нет Да Да Нет Нет Да Нет Нет
Process finished with exit code 0
|

"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\
Enter the two words: видео драйвер
Да Нет Да Да Нет
Process finished with exit code 0
|
```

Рисунок 6.15 - Примеры работы программы

ОТВЕТЫ НА ВОПРОСЫ

1. 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. Строки в апострофах и в кавычках, экранированные последовательности - служебные символы, "Сырые" строки, строки в тройных апострофах или кавычках.

3. Сложение, умножение, оператор принадлежности. Строковых функций в Python много, вот некоторые из них:

`chr()` – Преобразует целое число в символ

`ord()` – Преобразует символ в целое число

`len()` – Возвращает длину строки

`str()` – Изменяет тип объекта на `string`

4. 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”.

5. Если `s` это строка, выражение формы `s[m:n]` возвращает часть `s`, начинающуюся с позиции `m`, и до позиции `n`, но не включая позицию. Если пропустить первый индекс, срез начинается с начала строки. Аналогично, если опустить второй индекс `s[n:]`, срез длится от первого индекса до конца строки.

6. Более легкое представление в памяти.

7. `s.istitle()`

8. `if s1 in s2`

9. `s.find(<sub>)`.

10. `len(s)`

11. `s.count(<char>)`.

12. f-строки упрощают форматирование строк. Пример: `print(f" This is {name}, he is {age} years old")`

13. `string.find(<sub>[, <start>[, <end>]])`

14. `'Hello, {}'.format('Vasya')`

15. `string.isdigit()`

16. `'foo.bar.baz.qux'.rsplit(sep='.')` – пример разделения

17. `string.islower()`

18. `s[0].isupper()`

19. С точки зрения математической операции нельзя, можно лишь только вывести из без разделения друг от друга

20. `s[::-1]` – при помощи среза.

21. `'-'.join(<iterable>)`

22. К верхнему – `string.upper()`, к нижнему – `string.lower()`.

23. `s[0].upper()` `s[len(s) - 1].upper()`

24. `s.isupper()`

25. Если нужно сохранить символы, обозначающие конец слов.

26. `s.replace('что заменить', 'на что заменить')`

27. `string.endswith(<suffix>[, <start>[, <end>]])`, `str.startswith(prefix[, start[, end]])`

28. `s.isspace()`

29. Будет получена копия исходной строки в трёхкратном размере.

30. `s.title()`

31. `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>`

32. Когда нужен индекс последнего вхождения подстроки в строку.

