

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.3

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Турклиев В.Н. « » _____ 20_г.

Подпись студента _____

Работа защищена « » _____ 20_г.

**Проверил Воронкин Р.А. _____
(подпись)**

ВЫПОЛНЕНИЕ

Пример 1.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    s = input("Введите предложение: ")
    r = s.replace(' ', '_')
    print(f"Предложение после замены: {r}")

if __name__ == '__main__':
```

p1 ×

"C:\Users\Vova\Desktop\учеба\основы ии\lab6\lab6\

Введите предложение: лес озеро небо

Предложение после замены: лес_озеро_небо

Пример 2.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    word = input("Введите слово: ")
    idx = len(word) // 2
```

p2 ×

"C:\Users\Vova\Desktop\учеба\основы ии\lab6\lab6\

Введите слово: Владимир

Вламир

Пример 3.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-#!

import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))

    # Проверить требуемую длину.
    if len(s) >= n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)

    # Разделить предложение на слова.
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
        exit(1)

    # Количество пробелов для добавления.
    delta = n
    for word in words:
        delta -= len(word)

    # Количество пробелов на каждое слово.
    w, r = delta // (len(words) - 1), delta % (len(words) - 1)
```

```

# Сформировать список для хранения слов и пробелов.
lst = []

# Пронумеровать все слова в списке и перебрать их.
for i, word in enumerate(words):
    lst.append(word)

    # Если слово не является последним, добавить пробелы.
    if i < len(words) - 1:
        # Определить количество пробелов.
        width = w
        if r > 0:
            width += 1
            r -= 1

    # Добавить заданное количество пробелов в список.
    if width > 0:
        lst.append(' ' * width)

# Вывести новое предложение, объединив все элементы списка lst.
print(''.join(lst))

```

Введите предложение: *ночь покой равнина стол*
Введите длину: *30*
ночь покой равнина стол

Индивидуальное задание 1.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    s = str(input("Put some text: "))
    c = 0

    if (s[0] == ' ') and (s[len(s) - 1] == ' '):
        for i in s:
            if i == ' ':
                c += 1
        print(f"Количество слов в строке = {c - 1}")
    else:
        for i in s:
            if i == ' ':
                c += 1
        print(f"Количество слов в строке = {c + 1}")

if __name__ == '__main__': else

```

ind1 x

"C:\Users\Vova\Desktop\учеба\основы ии\lab6\lab6\pyProj\5

Put some text: *день ночь скрипка скалка*

Количество слов в строке = 4

Индивидуальное задание 2.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    s = str(input("Put some text: "))
    len = len(s)
    m = int(input("Put number of first letter you want to swap: "))
    n = int(input("Put number of second letter you want to swap: "))
    s1 = s[0:m]
    l1 = s[m]
    s2 = s[(m + 1):n]
    l2 = s[n]
    s = s1 + l2 + s2 + l1
    print(s)

if __name__ == '__main__':
```

ind2 ×

"C:\Users\Vova\Desktop\учеба\основы ии\lab6\lab6\pyProj\Scripts\python.exe
Put some text: **Флюгггехаймен**
Put number of first letter you want to swap: **5**
Put number of second letter you want to swap: **8**
Флюгггехг

Индивидуальное задание 3.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    s = str("килбайот")
    print(s)
    s1 = s[0:3]
    s2 = s[3:6]
    s3 = s[6]
    s4 = s[7]
    s = s1 + s3 + s2 + s4
    print(s)

if __name__ == '__main__':
```

ind3 ×

"C:\Users\Vova\Desktop\учеба\основы ии
килбайот
килобайт

Задание повышенной сложности:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    text = input("Put some text: ")
    result = ''
    for i in text.split():
        if text.count(i) == 1:
            result += i + ' '
    print(result)
```

if __name__ == '__main__' > for i in text.split() > if text.count(i) == 1

ind_hard ×

"C:\Users\Vova\Desktop\учеба\основы ии\lab6\lab6\pyProj\Scripts\python.e

Put some text: книга стол печать шкатулка час окно шкатулка книга дом

стол печать час окно дом

Ссылка – <https://github.com/vegas007gof/lab6>

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()`.
10. `len(s)`
11. `s.count()`.
12. f-строки упрощают форматирование строк. Пример: `print(f' This is {name}, he is {age} years old')`
13. `string.find([, [,]])`
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()`
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([, [,]])`, `str.startswith(prefix[, start[, end]])`
28. `s.isspace()`
29. Будет получена копия исходной строки в трёхкратном размере.
30. `s.title()`
31. `s.partition()` отделяет от `s` подстроку длиной от начала до первого вхождения `.`. Возвращаемое значение представляет собой кортеж из трех частей: Часть `s` до Разделитель Часть `s` после