

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

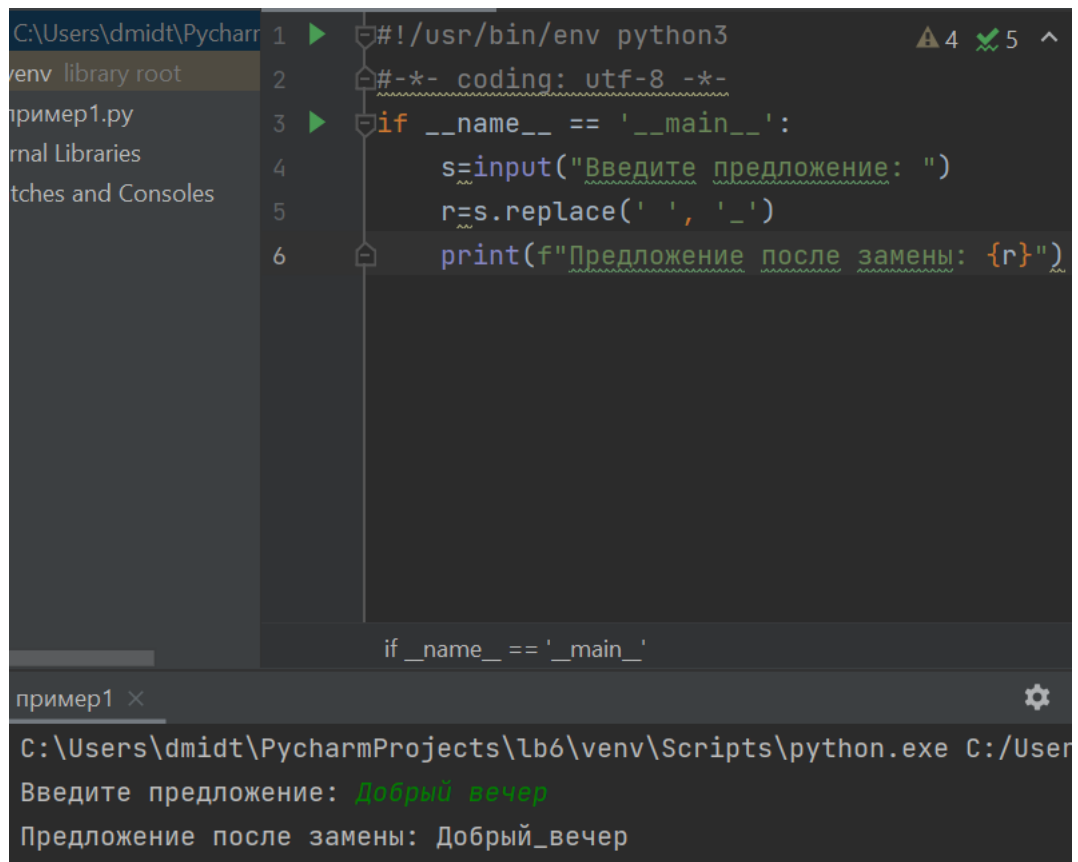
Отчет о лабораторной работе №6 по дисциплине
«Основы программной инженерии»

Выполнил:
Прокопов Дмитрий
Владиславович,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

Ход работы:

1-й пример



The screenshot shows the PyCharm IDE with a Python script named `пример1.py` open. The script is as follows:

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

Below the editor, the console output is visible:

```
C:\Users\dmidt\PycharmProjects\lb6\venv\Scripts\python.exe C:/User
Введите предложение: Добрый вечер
Предложение после замены: Добрый_вечер
```



2-й пример

```
C:\Users\dmidt\Pycharm 1 ▶ #!/usr/bin/env python3 8 2
venv library root 2 # -*- coding: utf-8 -*-
пример1.py 3 ▶ if __name__ == '__main__':
пример2.py 4 word=input("Введите слово: ")
ernal Libraries 5 idx=len(word)//2
atches and Consoles 6 if len(word)%2==1:
7 r=word[:idx]+word[idx+1:]
8 else:
9 r=word[:idx-1]+word[idx+1:]
10 print(r)

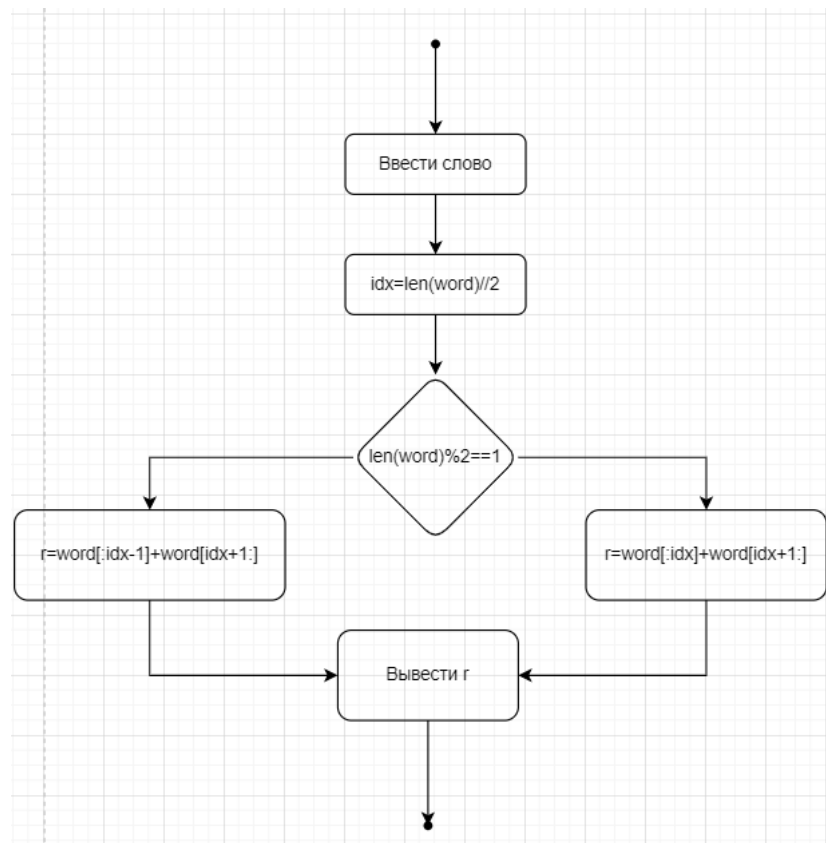
if __name__ == '__main__'
```

пример2 ×

C:\Users\dmidt\PycharmProjects\lb6\venv\Scripts\python.exe C:/U

Введите слово: пять

пъ



3-й пример

```

#!/usr/bin/env python3
#-*- coding: utf-8 -*-
import sys
if __name__ == '__main__':
    s=input("Введите предложение: ")
    n=int(input("Введите длину: "))
    if len(s)>=n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)
    words=s.split(' ')
    if len(words)<2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
        exit(1)
    delta=n
    for word in words:
        delta -=len(word)
    w, r=delta//(len(words)-1), delta%(len(words)-1)
    lst=[]

```

```

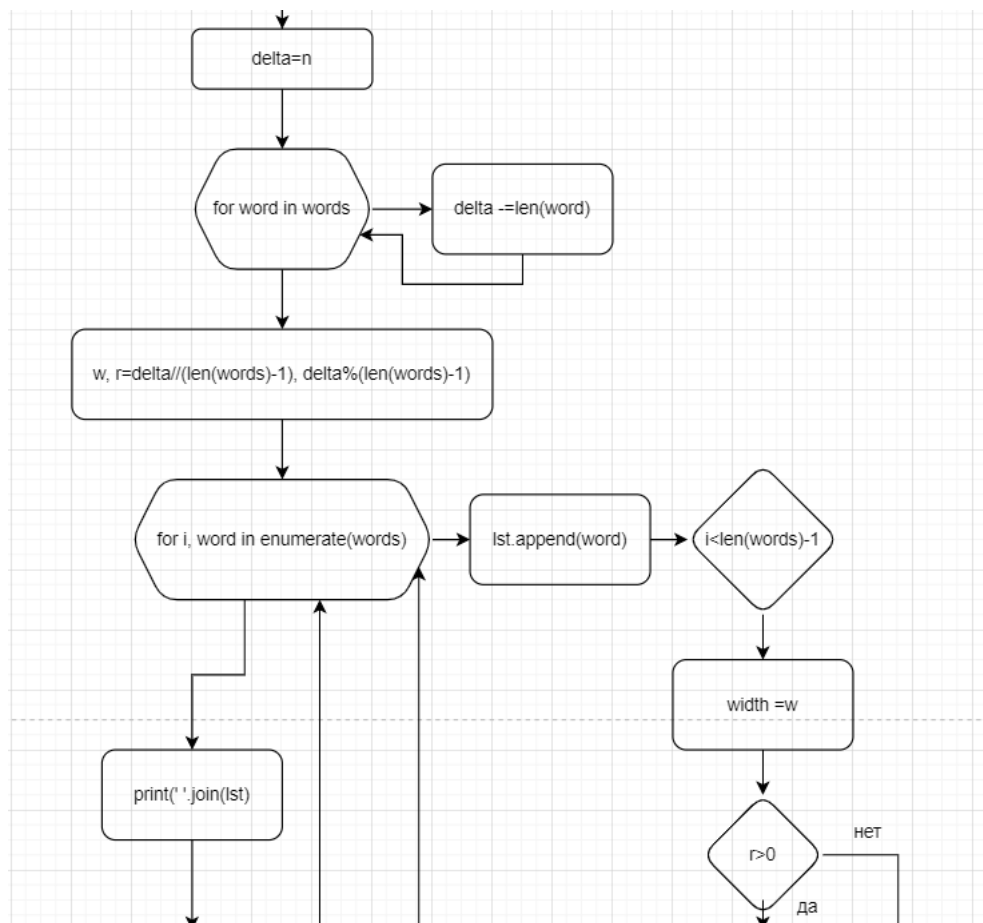
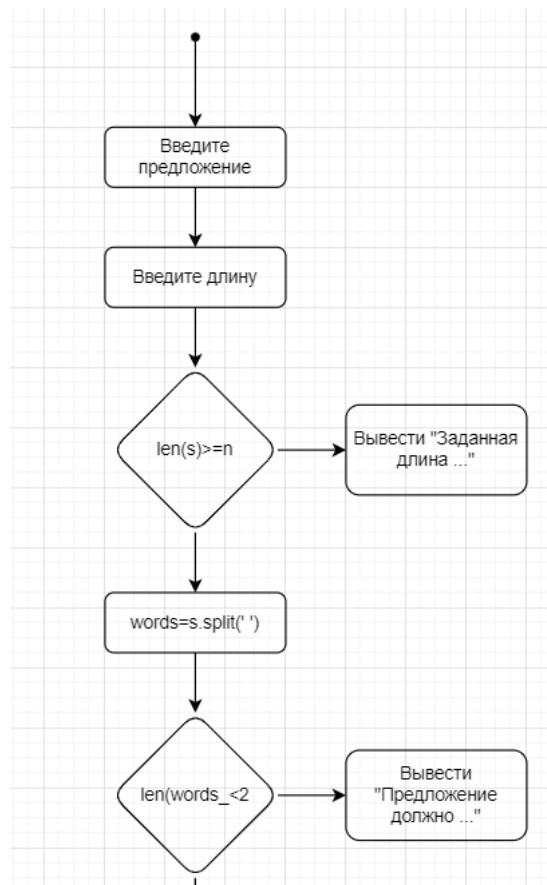
    for i, word in enumerate(words):
        lst.append(word)
        if i<len(words)-1:
            width=w
            if r>0:
                width +=1
                r -=1
            if width>0:
                lst.append(' ' * width)
    print(''.join(lst))

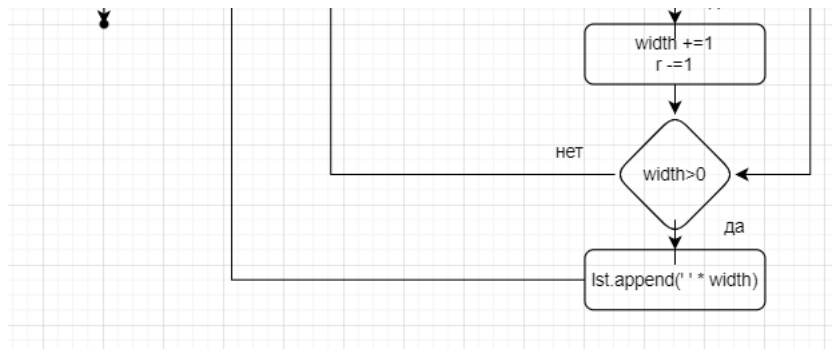
```

```

C:\Users\dmidt\PycharmProjects\lb6\venv\Scripts\p
Введите предложение: Добрый вечер ученики
Введите длину: 25
Добрый    вечер    ученики

```

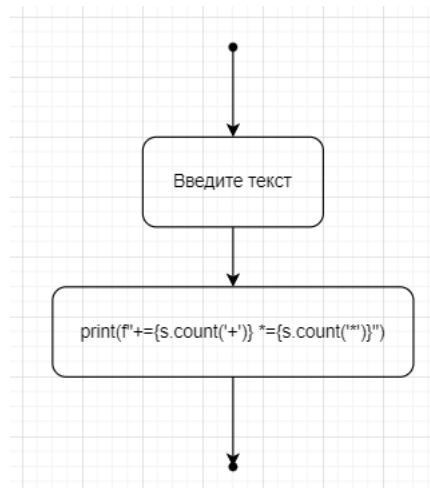




1 индивидуальное задание

```

C:\Users\dmidt\Pycharm 1
venv library root      2
nd1.py
пример1.py
ind1 x
C:\Users\dmidt\PycharmProjects\lb6\venv\Scripts\python.exe C:/Users/dm
Введите текст: gheglu+fegg+geg+gg*efeg* **grgh+grg+grgr++frgr+++feg*g
+=10 *=5
  
```



2 индивидуальное задание

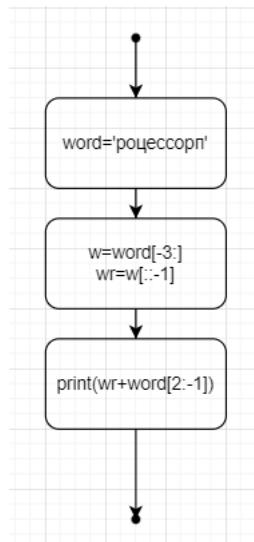
```
C:\Users\dmidt\Pycharm 1 ▶ #!/usr/bin/env python3
venv library root 2 # -*- coding: utf-8 -*-
nd1.py 3 ▶ if __name__ == '__main__':
nd2.py 4     s=input("Введите предложение: ")
пример1.py 5     r=s.replace('про', 'нет')
пример2.py 6     print(f"Предложение после замены: {r}")
пример3.py
ernal Libraries
atches and Consoles    if __name__ == '__main__'

ind2 ×
C:\Users\dmidt\PycharmProjects\lb6\venv\Scripts\python.exe C:/User
Введите предложение: про уаупро уаппроцкелпросыпро
Предложение после замены: нет уаунет уаппнетцкелнетсынет
```



3 индивидуальное задание

```
C:\Users\dmidt\Pycharm 1 word='процессорн'
venv library root 2 w=word[-3:]
nd1.py 3 wr=w[::-1]
nd2.py 4 print(wr + word[2:-1])
nd3.py 5
пример1.py
пример2.py
ind3 ×
C:\Users\dmidt\PycharmProjects\lb6\venv\Scripts\python.exe C:/U
процессор
```



Задание повышенной сложности

```

1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3  s=input("Введите предложение: ")
4  words=s.split()
5  print(min(words, key=len))
  
```

Введите предложение: дмдуашр кауэ ашыауо уоаг уао уагтф

уао



Ответы на вопросы:

1. Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации,

поэтому спомощью 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.isitle()`

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 тольковывести 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` доРазделитель Часть `s` после