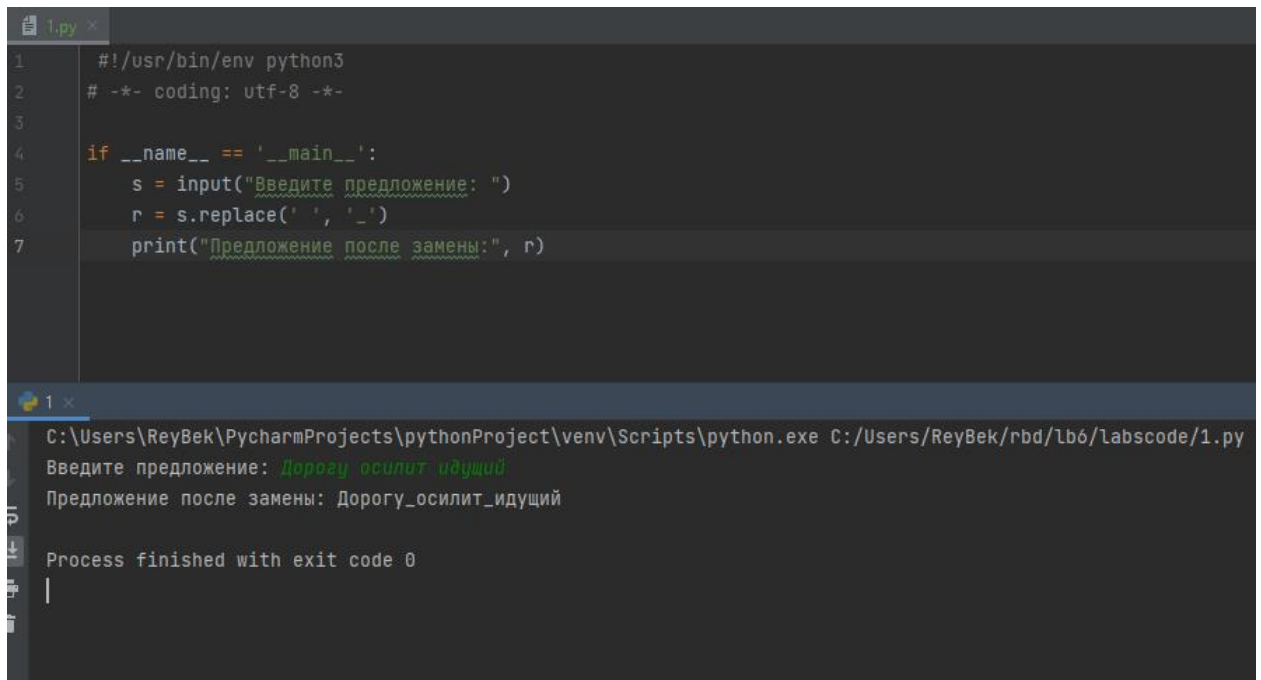


## ВЫПОЛНЕНИЕ

### 1) Пример 1

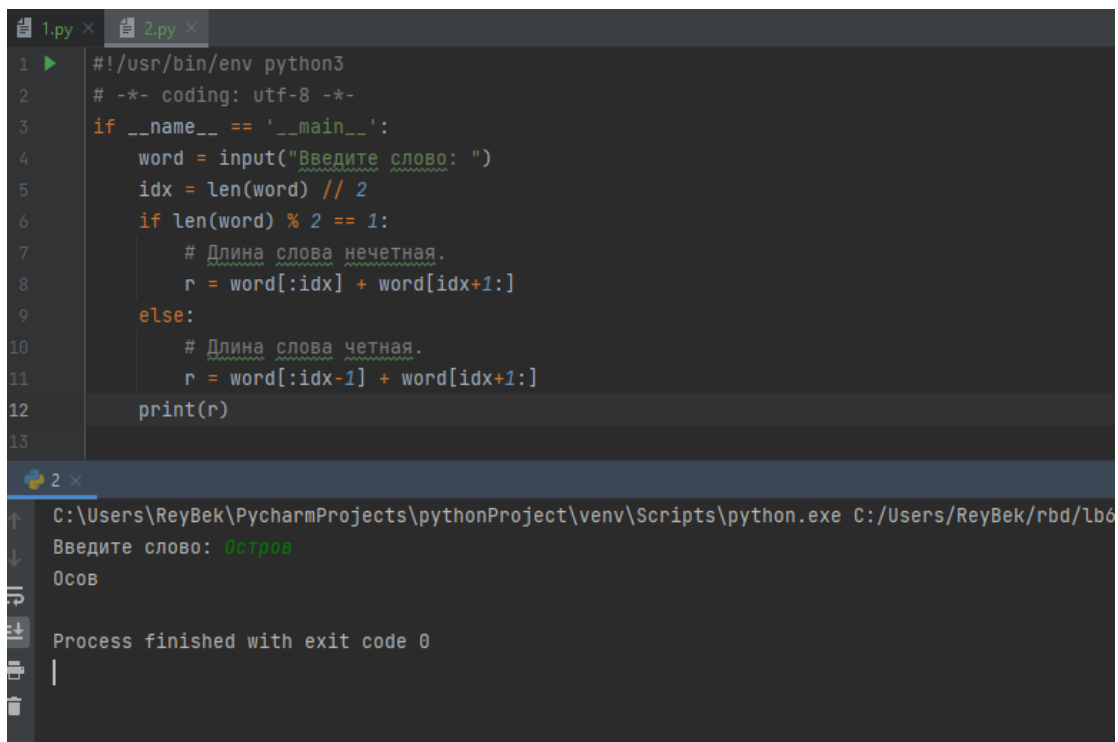


The screenshot shows a PyCharm IDE with a Python script named 1.py and its execution output. The script takes a sentence as input, replaces spaces with underscores, and prints the result. The execution shows the input "Дорогу осилит идущий" being processed into "Дорогу\_осилит\_идущий".

```
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("Предложение после замены:", r)
```

1 x  
C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb6/labscode/1.py  
Введите предложение: Дорогу осилит идущий  
Предложение после замены: Дорогу\_осилит\_идущий  
Process finished with exit code 0

### 2) Пример 2



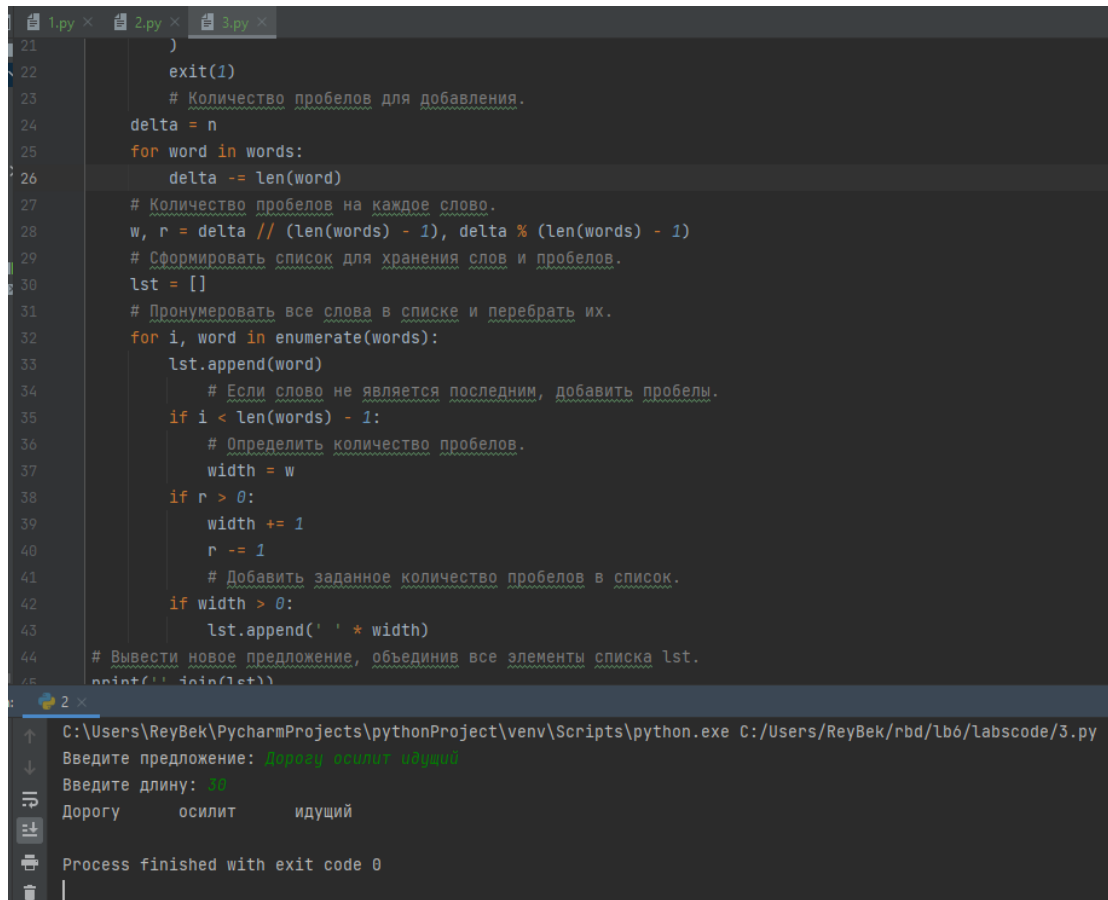
The screenshot shows a PyCharm IDE with a Python script named 2.py and its execution output. The script takes a word as input, calculates its length, and prints the word with its first and last characters swapped. The execution shows the input "Остров" being processed into "Осов".

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

2 x  
C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb6/labscode/2.py  
Введите слово: Остров  
Осов  
Process finished with exit code 0

Отчет №6 по дисциплине  
“Основы программной инженерии”  
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

### 3) Пример 3



The screenshot displays the PyCharm IDE interface. The top pane shows a Python script with the following code:

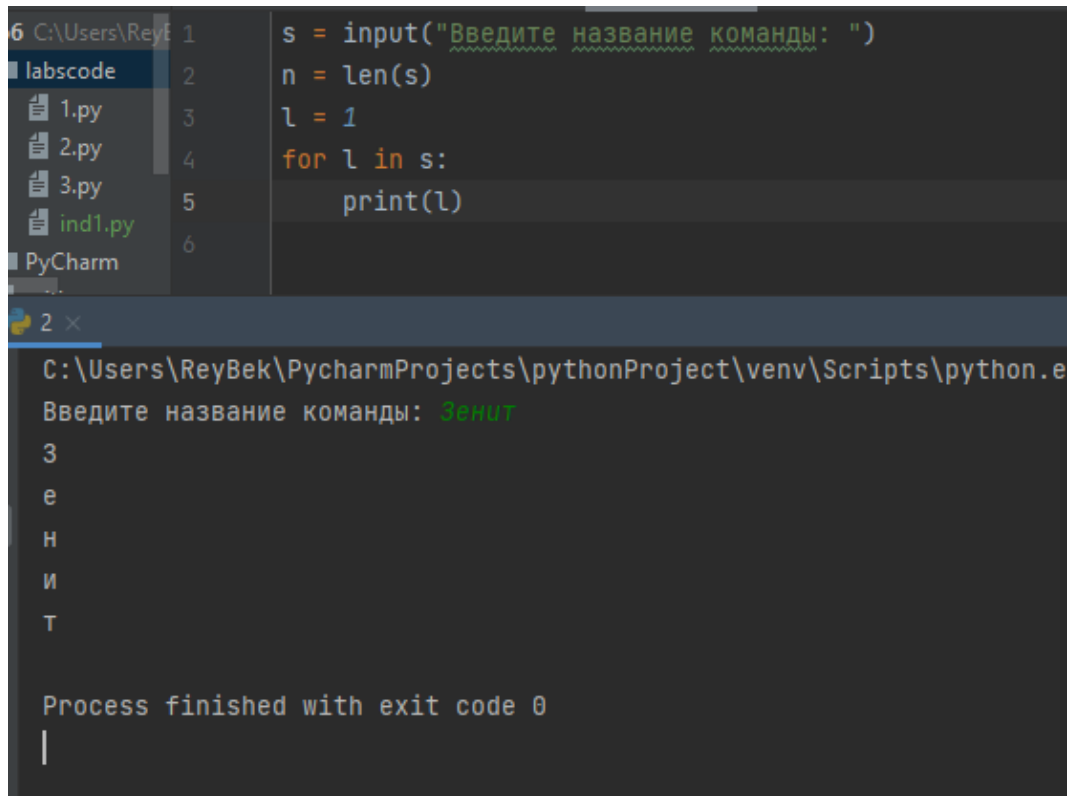
```
21 )
22     exit(1)
23     # Количество пробелов для добавления.
24     delta = n
25     for word in words:
26         delta -= len(word)
27
28     # Количество пробелов на каждое слово.
29     w, r = delta // (len(words) - 1), delta % (len(words) - 1)
30     # Сформировать список для хранения слов и пробелов.
31     lst = []
32     # Пронумеровать все слова в списке и перебрать их.
33     for i, word in enumerate(words):
34         lst.append(word)
35         # Если слово не является последним, добавить пробелы.
36         if i < len(words) - 1:
37             # Определить количество пробелов.
38             width = w
39             if r > 0:
40                 width += 1
41                 r -= 1
42             # Добавить заданное количество пробелов в список.
43             if width > 0:
44                 lst.append(' ' * width)
45
46 # Вывести новое предложение, объединив все элементы списка lst.
47 print(' '.join(lst))
```

The bottom pane shows the execution output:

```
C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb6/labscode/3.py
Введите предложение: Дорогу осилит идущий
Введите длину: 30
Дорогу    осилит    идущий
Process finished with exit code 0
```

### 4) Индивидуальное задание 1

Отчет №6 по дисциплине  
“Основы программной инженерии”  
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

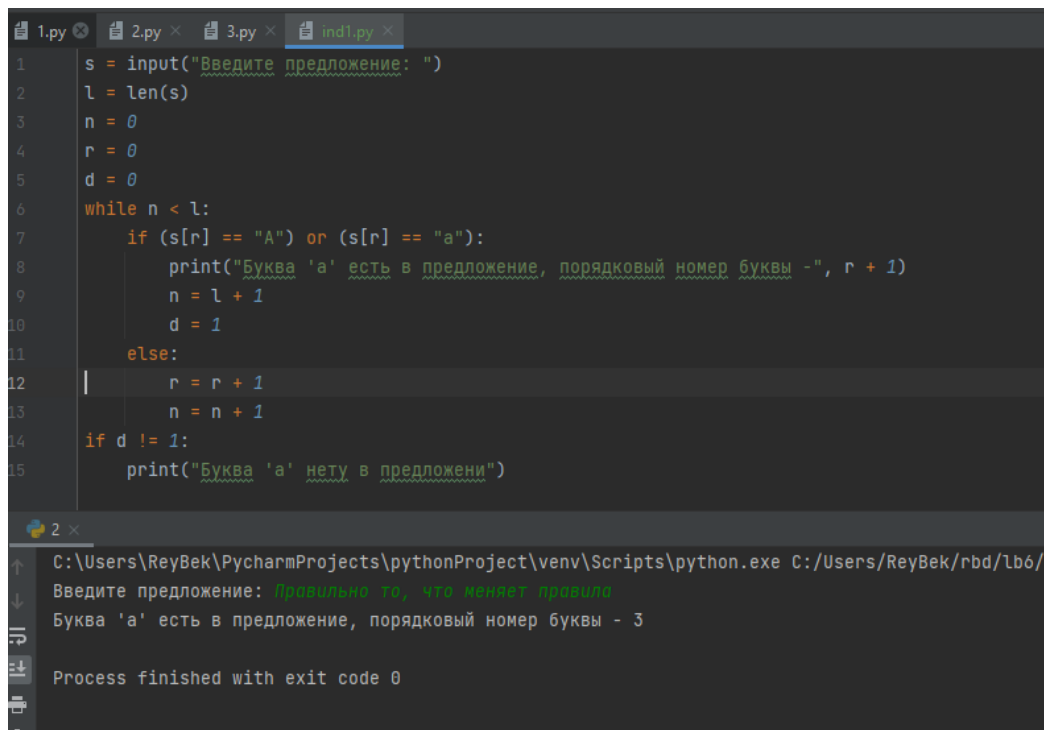


The screenshot shows the PyCharm IDE interface. On the left, a file explorer displays a project named 'labscod' containing files '1.py', '2.py', '3.py', and 'ind1.py'. The main editor window shows a Python script with the following code:

```
1 s = input("Введите название команды: ")
2 n = len(s)
3 l = 1
4 for l in s:
5     print(l)
6
```

Below the editor, the 'Run' tool window is open, showing the execution of the script. The command prompt displays the input 'Зенит' and the output 'З', 'е', 'н', 'и', 'т' on separate lines. The process finished with exit code 0.

## 5) Индивидуальное задание 2



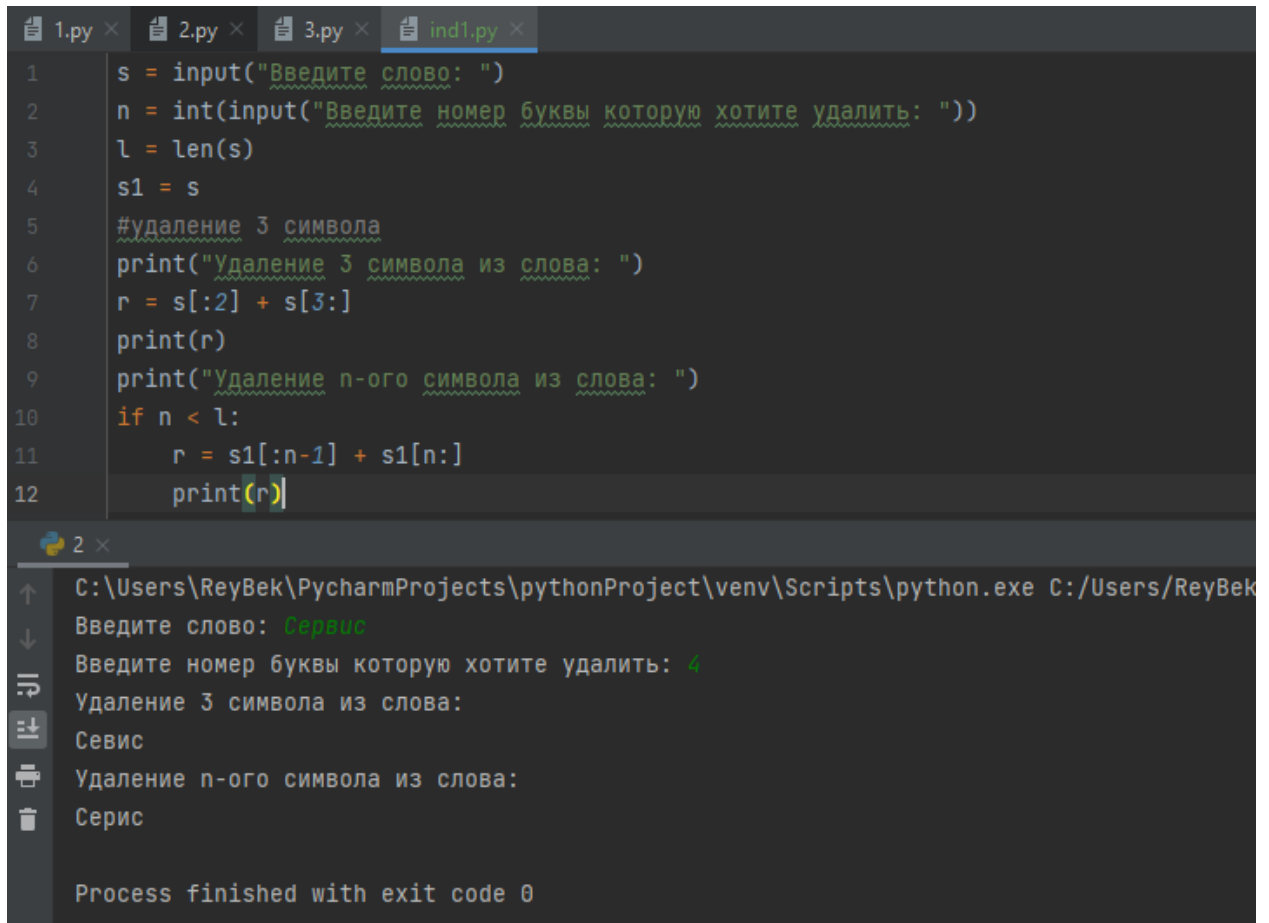
The screenshot shows the PyCharm IDE interface with multiple tabs open. The active tab is 'ind1.py', which contains the following Python code:

```
1 s = input("Введите предложение: ")
2 l = len(s)
3 n = 0
4 r = 0
5 d = 0
6 while n < l:
7     if (s[r] == "А") or (s[r] == "а"):
8         print("Буква 'а' есть в предложении, порядковый номер буквы -", r + 1)
9         n = l + 1
10        d = 1
11    else:
12        r = r + 1
13        n = n + 1
14 if d != 1:
15     print("Буква 'а' нету в предложени")
```

The 'Run' tool window shows the execution of the script. The command prompt displays the input 'Правильно то, что меняет правила' and the output 'Буква 'а' есть в предложении, порядковый номер буквы - 3'. The process finished with exit code 0.

## 6) Индивидуальное задание 3

Отчет №6 по дисциплине  
“Основы программной инженерии”  
Группа: ПИЖ-б-о-20-1, Бекишев М. М.



```
1  s = input("Введите слово: ")
2  n = int(input("Введите номер буквы которую хотите удалить: "))
3  l = len(s)
4  s1 = s
5  #удаление 3 символа
6  print("Удаление 3 символа из слова: ")
7  r = s[:2] + s[3:]
8  print(r)
9  print("Удаление n-ого символа из слова: ")
10 if n < l:
11     r = s1[:n-1] + s1[n:]
12     print(r)
```

2 x

C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek

Введите слово: Сервис

Введите номер буквы которую хотите удалить: 4

Удаление 3 символа из слова:

Сервис

Удаление n-ого символа из слова:

Сервис

Process finished with exit code 0

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации
2. `S = 'spam"s' S = "spam's"`
3. “+” – сложение строк, “\*” – умножение строк, “in” – принадлежность строки, “not in” – обратная принадлежность строки  
`Chr()` – преобразует число в символ `Ord()` – преобразует символ в число `Len()` – возвращает длину строки `Str()` – изменяет тип объекта на string
4. Доступ к отдельным символам в строке можно получить, указав имя строки, за которым следует число в квадратных скобках.

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 6. Строки неизменяемы, так как нет необходимости их менять

7. Сравнить строку со строкой с функцией title()

8. Использовать функцию find()

9. Find() возвращает первый индекс схождения

10.Функция len(s) возвращает длину строки

11.Функция count() возвращает количество точных вхождений подстроки в строку

12.F-строки это литерал отформатированной строки

13.В функцию count указать параметр start и end

14.для подстановки требуется аргумент, а в строке символы “{}”

15.s.isdigit() возвращает True когда строка s не пустая и все ее символы являются цифрами

16.string.partition() делит строку на основе разделителя.

17.Сравнить строку со строкой с функцией lower()

18.Сравнить первый символ строки с первым символом строки с функцией lower()

19.В python нельзя прибавить целое число к строке

20.Надо сменить порядок элементов строки

Отчет №6 по дисциплине  
“Основы программной инженерии”  
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

21.Командой '-'.join(['woof', 'woof'])

22.string.lower() преобразует все буквенные символы в строчные.

string.title() преобразует первые буквы всех слов в заглавные

23.string.lower() >>> string.swapcase()

24.преобразовать ссылку в список слов

25.метод replace() заменяет одно вхождение на заданное

26.string.endswith() определяет, заканчивается ли строка заданной

подстрокой string.startswith() определяет, начинается ли строка с

заданной подстроки

27.Надо подсчитать вхождение пробелов с длиной строки

28.Будет три одинаковые строки

29.string.title() преобразует первые буквы всех слов в заглавные

30.partition() где sep – разделяемый знак. Делит строку в список по

разделителю

31.Когда надо найти подстроку начиная с конца строки