

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

**Отчет по лабораторной работе №6 по дисциплине:
основы программной инженерии**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

1. Строки в тройных апострофах или кавычках

```
>>> C = '''Это очень большая
... строка, многострочный
... блок текста '''
>>> C
'Это очень большая \nстрока, многострочный\nблок текста '
>>> print(C)
Это очень большая
строка, многострочный
блок текста
```

2. Строковые операторы

```
>>> s = 'py'
>>> t = 'th'
>>> u = 'on'
>>> s+t
'pyth'
>>> s+t+u
'python'
>>> print('Привет, ' + 'Мир!')
Привет, Мир!
```

3. Выполнение умножения строк

```
>>> s = 'py.'
>>> s * 4
'py.py.py.py.'
>>> 4 * s
'py.py.py.py.'
>>> 'py' * -6
''
```

4. Оператор принадлежности подстроки in

```
>>> s = 'Python'
>>> s in 'I love Python'
True
>>> s in 'I love Java'
False
```

5. Выполнение операции not in

```
>>> 'z' not in 'abc'
True
>>> 'z' not in 'xyz'
False
```

6. Выполнение функции ord

```
>>> ord('a')
97
>>> ord('#')
35
```

7. Выполнение функции chr

```
>>> chr(97)
'a'
>>> chr(35)
'#'
>>> chr(8364)
'€'
>>> chr(8721)
'Σ'
```

8. Выполнение функции len(s)

```
>>> s = 'Простая строка.'  
>>> len(s)  
15
```

9. Возвращение строкового представления объекта

```
>>> str(49.2)  
'49.2'  
>>> str(3+4j)  
'(3+4j)'  
>>> str(3 + 29)  
'32'  
>>> str('py')  
'py'
```

10. Индексация строк

```
>>> s = 'foobar'  
>>> s[0]  
'f'  
>>> s[1]  
'o'  
>>> s[3]  
'b'  
>>> s[5]  
'r'  
>>> s[6]  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
IndexError: string index out of range
```

```

>>> s = 'foobar'
>>> s[-1]
'r'
>>> s[-2]
'a'
>>> len(s)
6
>>> s[-len(s)]
'f'
>>> s[-7]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range

```

11. Срезы строк

```

>>> s = 'python'
>>> s[2:5]
'tho'
>>> s = 'python'
>>> s[:4]
'pyth'
>>> s = 'python'
>>> s[2:]
'thon'
>>> s[2:len(s)]
'thon'
>>> s = 'python'
>>> s[:4] + s[4:]
'python'
>>> s[:4] + s[4:] == s
True

```

```
>>> s = 'python'
>>> t = s[:]
>>> id(s)
2288806204976
>>> id(t)
2288806204976
>>> s is t
True
>>> s[2:2]
''
>>> s[4:2]
''
>>> s = 'python'
>>> s[-5:-2]
'yth'
>>> s[1:4]
'yth'
>>> s[-5:-2] == s[1:4]
True
```

12. Шаг для среза строки

```
>>> s = 'foobar'
>>> s[0:6:2]
'foa'
>>> s[1:6:2]
'obr'
>>> s = '12345' * 5
>>> s
'1234512345123451234512345'
>>> s[::5]
'11111'
>>> s[4::5]
'55555'
>>> s = 'python'
>>> s[5:0:-2]
'nhy'
>>> s = 'Если так говорит товарищ Наполеон, значит, так оно и есть.'
>>> s[::-1]
'.ьтсе и оно кат ,тичанз ,ноелопан щиравог тировог кат илсЕ'
```

13. Форматирование строки

```
>>> 'Hello,{ }!' .format('Vasya')
'Hello,Vasya!'
>>> '{0}, {1}, {2}' .format('a', 'b', 'c')
'a, b, c'
>>> '{2}, {1}, {0}' .format(*'abc')
'c, b, a'
>>> 'Coordinates: {latitude}, {longitude}'.format(latitude='37.24N',
... longitude='-115.81W')
'Coordinates: 37.24N, -115.81W'
>>> coord = {'latitude': '37.24N', 'longitude': '-115.81W'}
>>> 'Coordinates: {latitude}, {longitude}'.format(**coord)
'Coordinates: 37.24N, -115.81W'
>>> "Units destroyed: {players[0]}".format(players = [1, 2, 3])
'Units destroyed: 1'
```

```

>>> 'X: {0[0]}; Y: {0[1]}'.format(coord)
'X: 3; Y: 5'
>>> '{:>30}'.format('right aligned')
'                right aligned'
>>> '{:^30}'.format('centered')
'                centered                '
>>> '{:*^30}'.format('centered') # use '*' as a fill char
'*****centered*****'
>>> '{:+f}; {:+f}'.format(3.14, -3.14) # show it always
'+3.140000; -3.140000'

```

14. Форматирование с помощью f-строк

```

>>> n = 20
>>> m = 25
>>> prod = n * m
>>> print('Произведение', n, 'на', m, 'равно', prod)
Произведение 20 на 25 равно 500
>>> n = 20
>>> m = 25
>>> prod = n * m
>>> print(f'Произведение {n} на {m} равно {prod}')
Произведение 20 на 25 равно 500
>>> var = 'Гав'
>>> print(f'Dog said {var}!')
Dog said Гав!

```


15. Изменение строк

```
>>> s = 'python'
>>> s[3] = 't'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
>>> s = s[:3] + 't' + s[4:]
>>> s
'pytton'
>>> s = 'python'
>>> s = s.replace('h', 't')
>>> s
'pytton'
```

16. Изменение регистра строки

```
>>> s = 'everyTHing yoU Can IMaGine is rEaL'
>>> s.capitalize()
'Everything you can imagine is real'
>>> s = 'follow us @PYTHON'
>>> s.capitalize()
'Follow us @python'
>>> 'everyTHing yoU Can IMaGine is rEaL'.lower()
'everything you can imagine is real'
>>> 'everyTHing yoU Can IMaGine is rEaL'.swapcase()
'EVERYthING YOu cAN imAgINE IS ReaL'
>>> 'the sun also rises'.title()
'The Sun Also Rises'
```

17. Поиск и замена подстроки в строке

```
>>> 'foo goo moo'.count('oo', 0, 8)
2
>>> 'python'.endswith('on')
True
>>> 'python'.endswith('or')
False
>>> 'python'.endswith('yt', 0, 4)
False
>>> 'python'.endswith('yt', 2, 4)
False
>>> 'Follow Us @Python'.find('Us')
7
>>> 'Follow Us @Python'.find('Us', 4)
7
>>> 'Follow Us @Python'.find('Us', 4, 7)
-1
>>> 'Follow Us @Python'.index('you')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: substring not found
>>> 'Follow Us @Python'.rfind('o')
15
>>> 'Follow Us @Python'.rfind('a')
-1
>>> 'Follow Us @Python'.rfind('Us', 0, 14)
7
>>> 'Follow Us @Python'.rfind('Us', 9, 14)
-1
```

18. Классификация строк

```
>>> 'abc123'.isalpha()
False
>>> '123'.isdigit()
True
>>> '123abc'.isdigit()
False
>>> 'foo32'.isidentifier()
True
>>> '32foo'.isidentifier()
False
>>> from keyword import iskeyword
>>> iskeyword('and')
True
>>> 'abc'.islower()
True
>>> 'abc1$d'.islower()
True
>>> 'Abc1$D'.islower()
False
>>> 'a\tb'.isprintable()
False
>>> 'a b'.isprintable()
True
>>> '\t\n'.isspace()
True
>>> ' a '.isspace()
False
>>> '\f\u2005\r'.isspace()
True
```

19. Выравнивание строк, отступы

```
>>> 'py'.center(10)
'   py   '
>>> 'py'.center(10, '-')
'----py----'
>>> 'a\tb\tc'.expandtabs()
'a      b      c'
>>> 'aaa\tbbb\tc'.expandtabs()
'aaa    bbb    c'
>>> 'a\tb\tc'.expandtabs(4)
'a  b  c'
>>> 'aaa\tbbb\tc'.expandtabs(tabsize=4)
'aaa bbb c'
>>> 'python'.ljust(10)
'python   '
>>> 'python'.ljust(10, '-')
'python----'
>>> 'python'.ljust(2)
'python'
>>> ' foo bar baz '.lstrip()
'foo bar baz '
>>> '\t\nfoo\t\nbar\t\nbaz'.lstrip()
'foo\t\nbar\t\nbaz'
>>> 'I hate python! I hate python! I hate python!'.replace('hate', 'love')
'I love python! I love python! I love python!'
>>> 'python'.rjust(10)
'   python'
>>> 'python'.rjust(10, '-')
'----python'
>>> '42'.zfill(5)
'00042'
```

20. Методы преобразования строки в список

```
>>> ','.join(['foo', 'bar', 'baz', 'qux'])
'foo, bar, baz, qux'
>>> list('corge')
['c', 'o', 'r', 'g', 'e']
>>> ':'.join('corge')
'c:o:r:g:e'
>>> '---'.join(['foo', str(23), 'bar'])
'foo---23---bar'
>>> 'foo.bar'.partition('.')
('foo', '.', 'bar')
>>> 'foo bar baz qux'.rsplit()
['foo', 'bar', 'baz', 'qux']
>>> 'foo.bar.baz.qux'.rsplit(sep='.')
['foo', 'bar', 'baz', 'qux']
>>> 'foo\t\t\tbar'.rsplit()
['foo', 'bar']
>>> 'foo\nbar\r\nbaz\fqux\u2028quux'.splitlines()
['foo', 'bar', 'baz', 'qux', 'quux']
```

21.Выполнение 1 примера

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

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

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/modul1.py

Введите предложение: *Предложение после замены*

Предложение после замены: Предложение_после_замены

22.Выполнение 2 примера

```
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          r = word[:idx] + word[idx + 1:]
10     else:
11         r = word[:idx - 1] + word[idx + 1:]
12
13     print(r)
14
15
16
17
18
19     if __name__ == '__main__':
```

in: modul2 ×

↑ C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/modul2.py

↓ Введите слово: **Самоменеджмент**

Ит Самомеджмент

23.Выполнение примера 3

```
19         "Предложение должно содержать несколько слов",
20         file=sys.stderr
21     )
22     exit(1)
23
24     delta = n
25     for word in words:
26         delta -= len(word)
27
28     w, r = delta // (len(words) - 1), delta % (len(words) - 1)
29
30     lst = []
31
32     for i, word in enumerate(words):
33         lst.append(word)
34
35     if i < len(words) - 1:
36         width = w
37         if r > 0:
38             width += 1
39
40     if __name__ == '__main__':
41         for i, word in enumerate(words)
```

run: modul3 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/modul3.py

Введите предложение: *привет как дела*

Введите длину: *20*

привет как дела

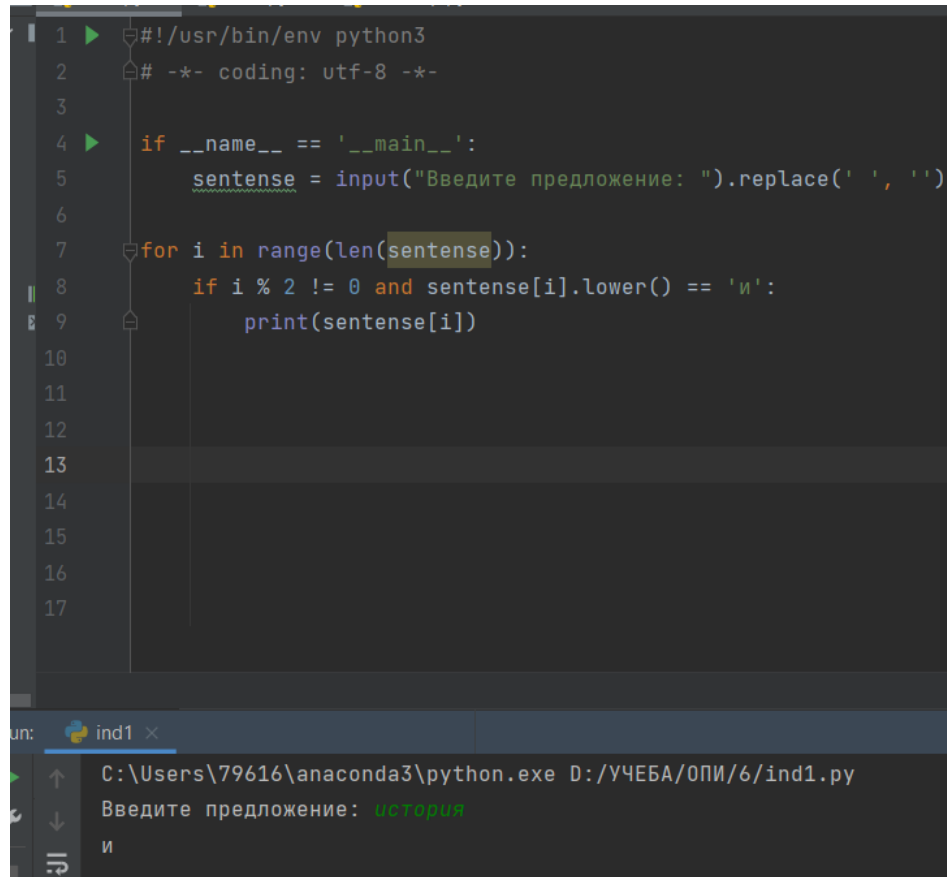
Выполнение индивидуальных заданий.
Вариант 13.

Задание 1: Дано предложение. Вывести столбиком все буквы И, стоящие на четных местах

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

if __name__ == '__main__':
    sentence = input("Введите предложение: ").replace(' ', '')

    for i in range(len(sentence)):
        if i % 2 != 0 and sentence[i].lower() == 'и':
            print(sentence[i])
```



```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == '__main__':
5      sentence = input("Введите предложение: ").replace(' ', '')
6
7      for i in range(len(sentence)):
8          if i % 2 != 0 and sentence[i].lower() == 'и':
9              print(sentence[i])
10
11
12
13
14
15
16
17
```

un: ind1 ×

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/ind1.py

Введите предложение: история

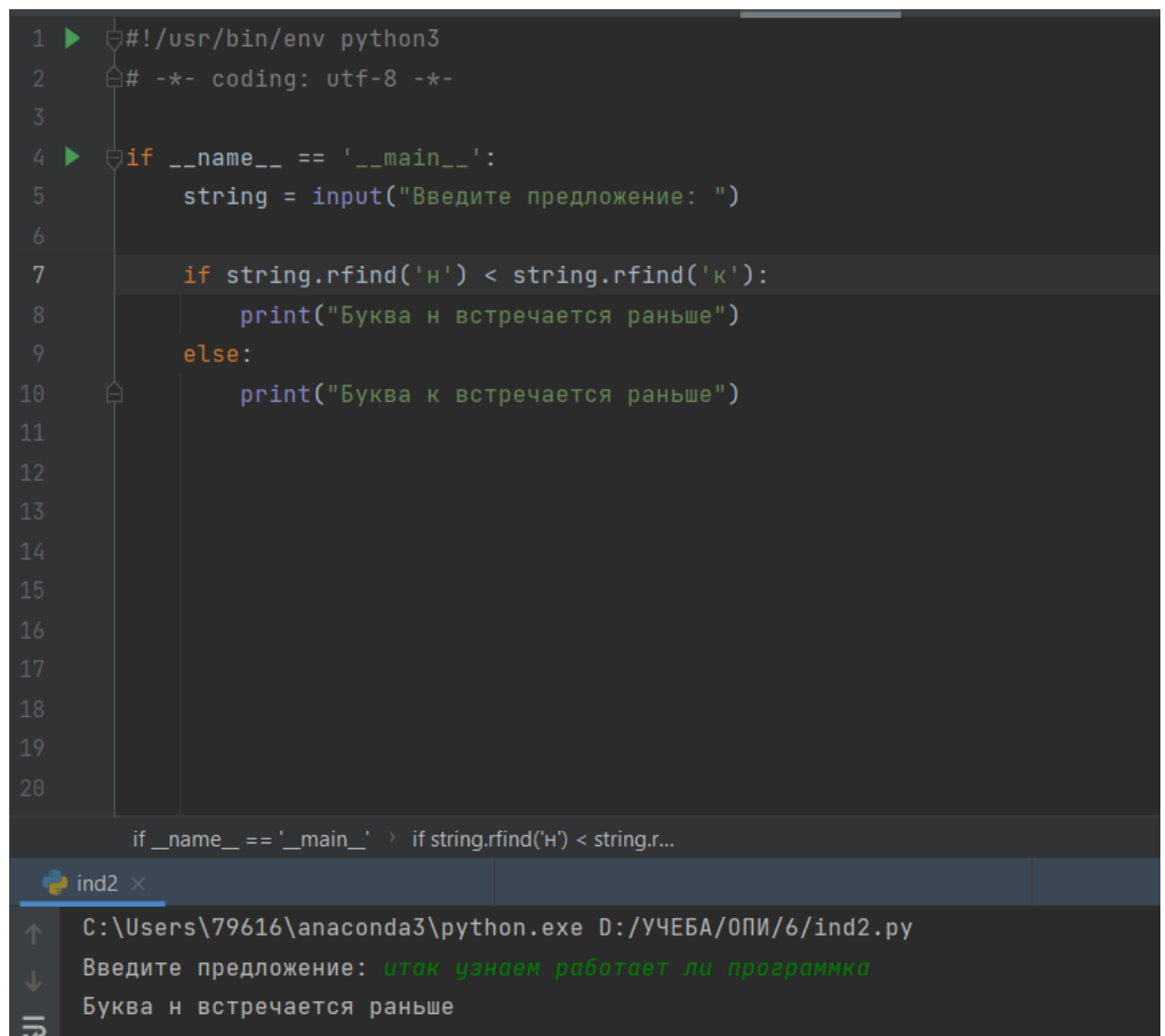
и

Задание 2: Дано предложение. Определить, какая из букв н или к встречается в ней раньше при просмотре слева направо

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

if __name__ == '__main__':
    string = input("Введите предложение: ")

    if string.rfind('н') < string.rfind('к'):
        print("Буква н встречается раньше")
    else:
        print("Буква к встречается раньше")
```



The screenshot shows a code editor with a Python script and its execution output. The script is the same as the one in the previous block. The execution output shows the program running and printing the result.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     string = input("Введите предложение: ")
6
7     if string.rfind('н') < string.rfind('к'):
8         print("Буква н встречается раньше")
9     else:
10        print("Буква к встречается раньше")
11
12
13
14
15
16
17
18
19
20
```

if __name__ == '__main__' > if string.rfind('н') < string.r...

ind2 ×

↑ C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/ind2.py

↓ Введите предложение: *итак узнаем работает ли программка*

Буква н встречается раньше

Задание 3: Дано ошибочно написанное слово рпроцессо. Путем перемещения его букв получить слов процессор.

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

if __name__ == '__main__':
    string1 = "рпроцессо"
    string1 = string1[1:4] + string1[4:9] + string1[0]
    print(string1)
```

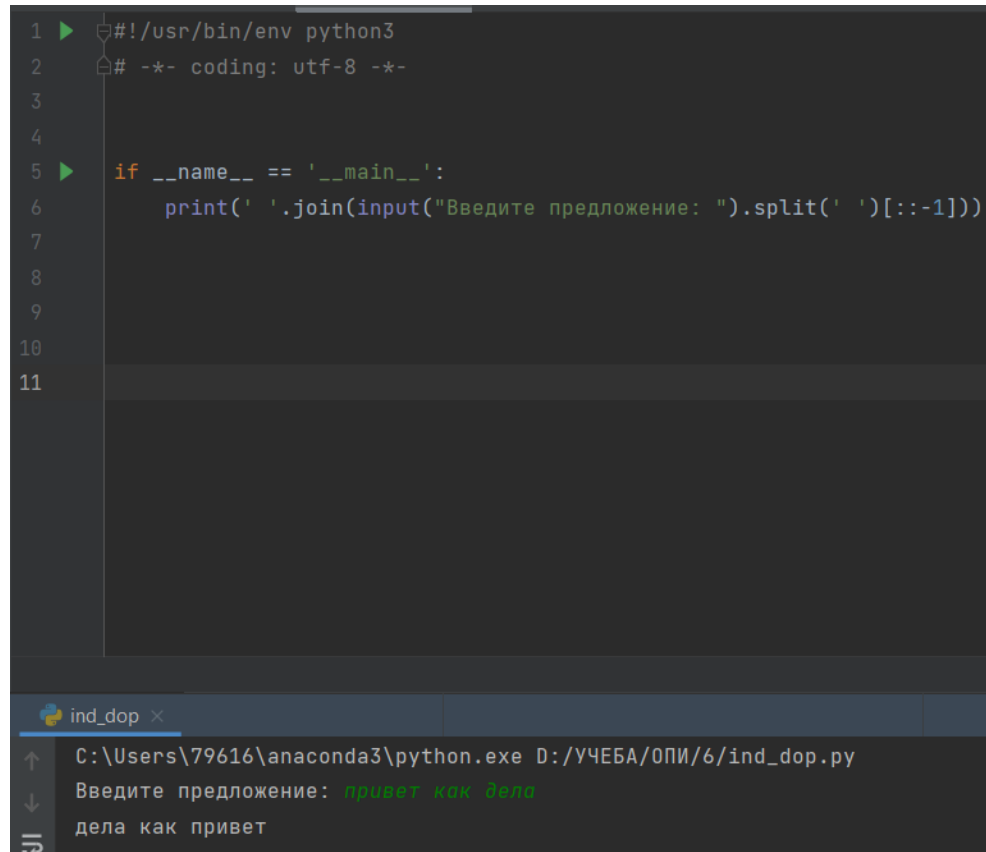
```
1 ▶ 1#!/usr/bin/env python3
2 2# -*- coding: utf-8 -*-
3
4 ▶ 4if __name__ == '__main__':
5     string1 = "рпроцессо"
6     string1 = string1[1:4] + string1[4:9] + string1[0]
7     print(string1)
8

ind3 ×
C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/ind3.py
↑
↓
процессор
```

Задание повышенной сложности. Дано предложение. Напечатать его в обратном порядке слов.

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

if __name__ == '__main__':
    print(' '.join(input("Введите предложение: ").split(' ')[::-1]))
```



The screenshot shows a code editor with a Python script and a terminal window below it. The script is the same as the one in the previous block. The terminal window, titled 'ind_dop', shows the command prompt 'C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/ind_dop.py' and the program's execution. It prompts 'Введите предложение: ' and shows the input 'привет как дела' in green. The output is 'дела как привет'.

```
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4
5 ▶   if __name__ == '__main__':
6       print(' '.join(input("Введите предложение: ").split(' ')[::-1]))
7
8
9
10
11
```

ind_dop ×

↑ C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/6/ind_dop.py

↓ Введите предложение: *привет как дела*

⏮ дела как привет