

Федеральное государственное автономное образовательное учреждение
высшего образования «Северо-Кавказский федеральный университет»
Институт цифрового развития

ОТЧЕТ ПО ДИСЦИПЛИНЕ
ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Лабораторная работа №6
Работа со строками в языке Python

Выполнила:

Мирзаева Камилла Мирзаевна
2 курс, группа ПИЖ-б-о-20-1

Принял:

Воронкин Роман Александрович

Ставрополь, 2021 г.

Ход работы:

```
>>> 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
```

Рисунок 1 - Строки в тройных апострофах или кавычках

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

Рисунок 2 - Строковые операторы

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

Рисунок 3 - Выполнение умножения строк

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

Рисунок 4 - Оператор принадлежности подстроки in

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

Рисунок 5 - Выполнение операции not in

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

Рисунок 6 - Выполнение функции ord

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

Рисунок 7 - Выполнение функции chr

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

Рисунок 8 - Выполнение функции len(s)

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

Рисунок 9 - Возвращение строкового представления объекта

```
>>> 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
```

Рисунок 10 - Индексация строк

```
>>> 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
```

Рисунок 11 - Индексация 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
```

Рисунок 12 - Срезы строк

```

>>> 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

```

Рисунок 13 - Срезы строк

```

>>> s = 'foobar'
>>> s[0:6:2]
'foa'
>>> s[1:6:2]
'obr'
>>> s = '12345' * 5
>>> s
'123451234512345123451234512345'
>>> 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 кат илсЕ'

```

Рисунок 14 - Шаг для среза 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'

```

Рисунок 15 - Форматирование строки

```

>>> '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'

```

Рисунок 16 - Форматирование строки

```

>>> 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!

```

Рисунок 17 - Форматирование с помощью f-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'

```

Рисунок 18 - Изменение строк


```
>>> 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'
```

Рисунок 19 - Изменение регистра строки

```
>>> '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
```

Рисунок 20 - Поиск и замена подстроки в строке

```
>>> '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
```

Рисунок 21 - Классификация строк

```

>>> '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'

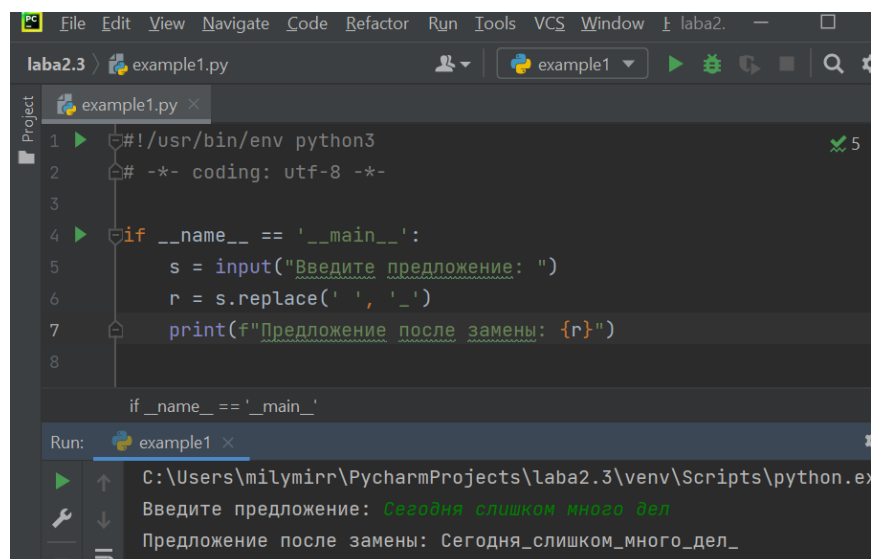
```

Рисунок 22 - Выравнивание строк, отступы

```
>>> ', '.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']
```

Рисунок 23 - Методы преобразования строки в список

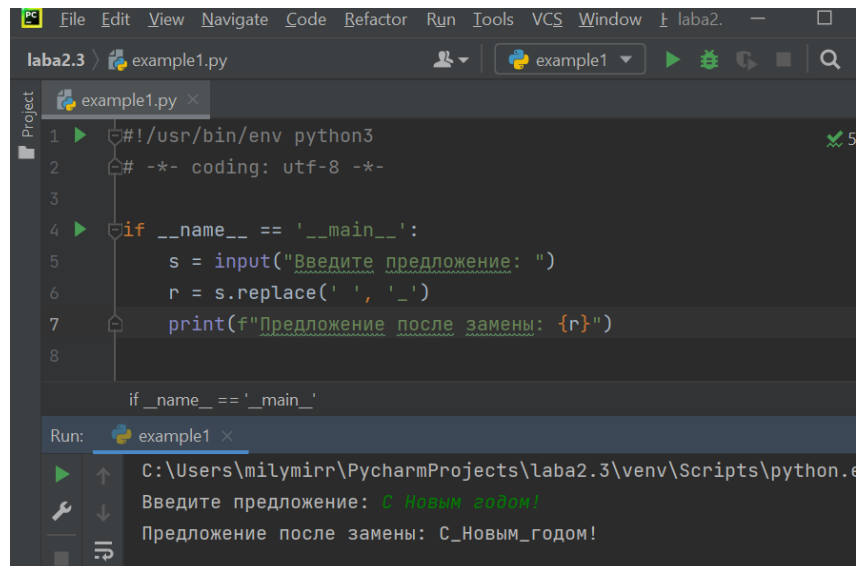
Пример №1



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and a search icon. The main editor window displays a file named 'example1.py' with the following code:

```
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}")
8
```

Below the editor, the 'Run' tab is active, showing the execution of 'example1'. The command line shows the path to the Python interpreter: 'C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe'. The input prompt is 'Введите предложение: Сегодня слишком много дел'. The output is 'Предложение после замены: Сегодня_слишком_много_дел_'.



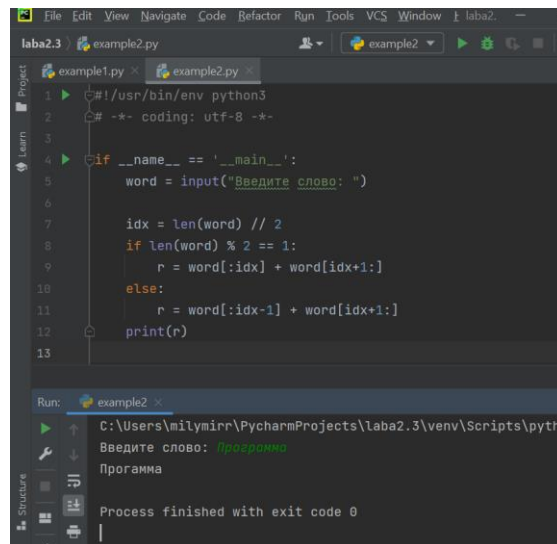
The image shows the PyCharm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, and Window. The project name is 'laba2.3'. The file 'example1.py' is open in the editor. The code is as follows:

```
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}")
8
9 if __name__ == '__main__':
```

Below the editor, the 'Run' console shows the execution of 'example1'. The output is:

```
C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe
Введите предложение: С Новым годом!
Предложение после замены: С_Новым_годом!
```

Пример №2

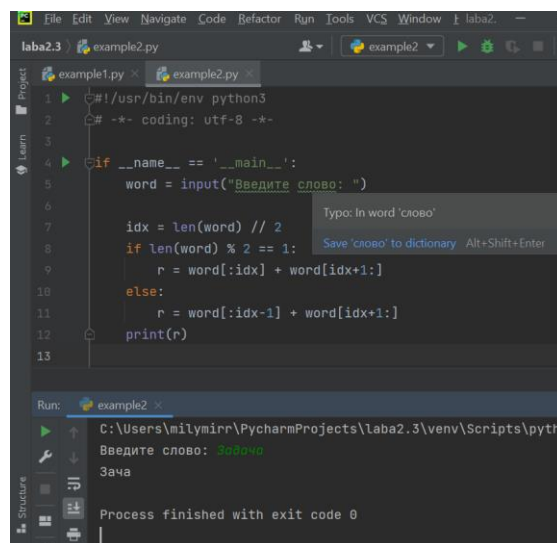


The image shows the PyCharm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, and Window. The project name is 'laba2.3'. The file 'example2.py' is open in the editor. The code is as follows:

```
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     print(r)
13
```

Below the editor, the 'Run' console shows the execution of 'example2'. The output is:

```
C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe
Введите слово: Программа
Программа
Process finished with exit code 0
```



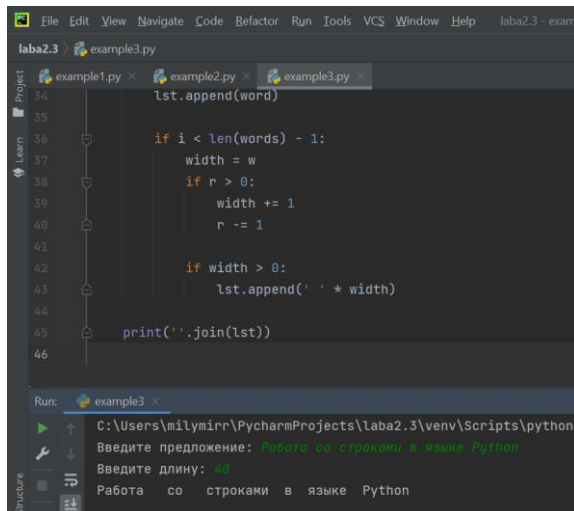
The image shows the PyCharm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, and Window. The project name is 'laba2.3'. The file 'example2.py' is open in the editor. The code is as follows:

```
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     print(r)
13
```

A tooltip is visible over the variable 'word' in line 5, showing the text: 'Тип: In word 'слово''. Below the editor, the 'Run' console shows the execution of 'example2'. The output is:

```
C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe
Введите слово: Задача
Задача
Process finished with exit code 0
```

Пример №3



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help lab2.3 - exam...
lab2.3 example3.py
34 lst.append(word)
35
36
37 if i < len(words) - 1:
38     width = w
39     if r > 0:
40         width += 1
41         r -= 1
42
43     if width > 0:
44         lst.append(' ' * width)
45
46 print(' '.join(lst))

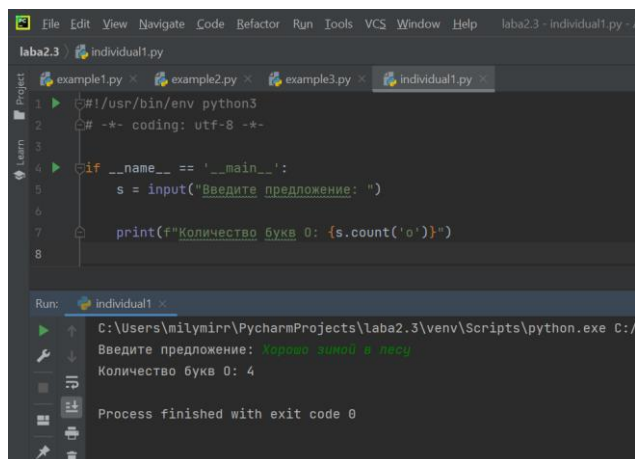
Run: example3
C:\Users\milymirr\PycharmProjects\lab2.3\venv\Scripts\python.
Введите предложение: Работа со строками в языке Python
Введите длину: 40
Работа со строками в языке Python
```

Индивидуальные задания.

Вариант №15.

Задание №1.

Дано предложение. Определить число букв о в нем.

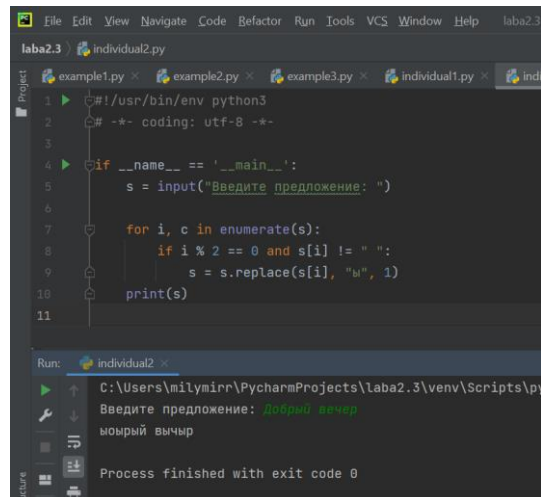


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help lab2.3 - individual1.py - F...
lab2.3 individual1.py
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     s = input("Введите предложение: ")
6
7     print(f"Количество букв о: {s.count('o')}")
8

Run: individual1
C:\Users\milymirr\PycharmProjects\lab2.3\venv\Scripts\python.exe C:/
Введите предложение: хорошо зимой в лесу
Количество букв о: 4
Process finished with exit code 0
```

Задание №2.

Дано предложение. Все его символы, стоящие на четных местах, заменить буквой ы.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     s = input("Введите предложение: ")
6
7     for i, c in enumerate(s):
8         if i % 2 == 0 and s[i] != " ":
9             s = s.replace(s[i], "ы", 1)
10    print(s)
```

Run: individual2 ×

C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe

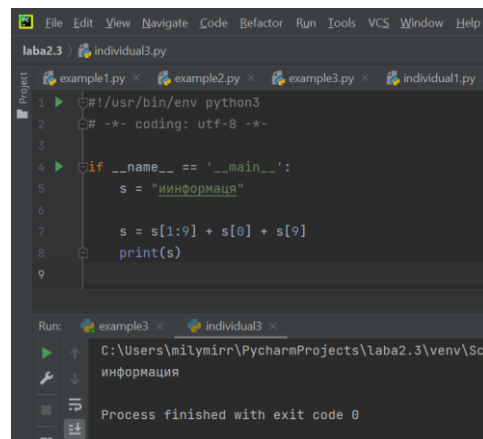
Введите предложение: Добрый вечер

Добрый вычвр

Process finished with exit code 0

Задание №3

Дано ошибочно написанное слово иинформация. Путем перемещения его букв получить слово информация.



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

Run: example3 × individual3 ×

C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe

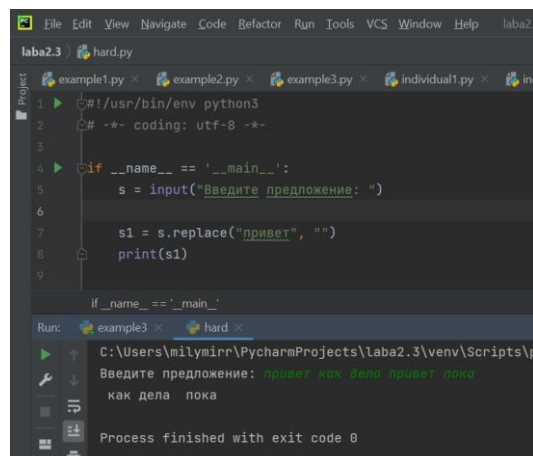
иинформация

информация

Process finished with exit code 0

Здание повышенной сложности:

Дано предложение. Напечатать все его слова, отличные от слова привет.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     s = input("Введите предложение: ")
6
7     s1 = s.replace("привет", "")
8     print(s1)
```

Run: example3 × hard ×

C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe C:\Users\milymirr\PycharmProjects\laba2.3\venv\Scripts\python.exe

Введите предложение: привет как дело привет пока

как дела пока

Process finished with exit code 0