

**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:  
Гребе Владислав  
Александрович,  
2 курс, группа ПИЖ-б-о-20-1,

Проверил:  
Доцент кафедры  
прикладной математики и  
компьютерной безопасности,  
Воронкин Р.А.

Отчет защищен с оценкой\_\_\_\_\_Дата защиты\_\_\_\_\_

Ставрополь, 2021 г.

## ВЫПОЛНЕНИЕ:

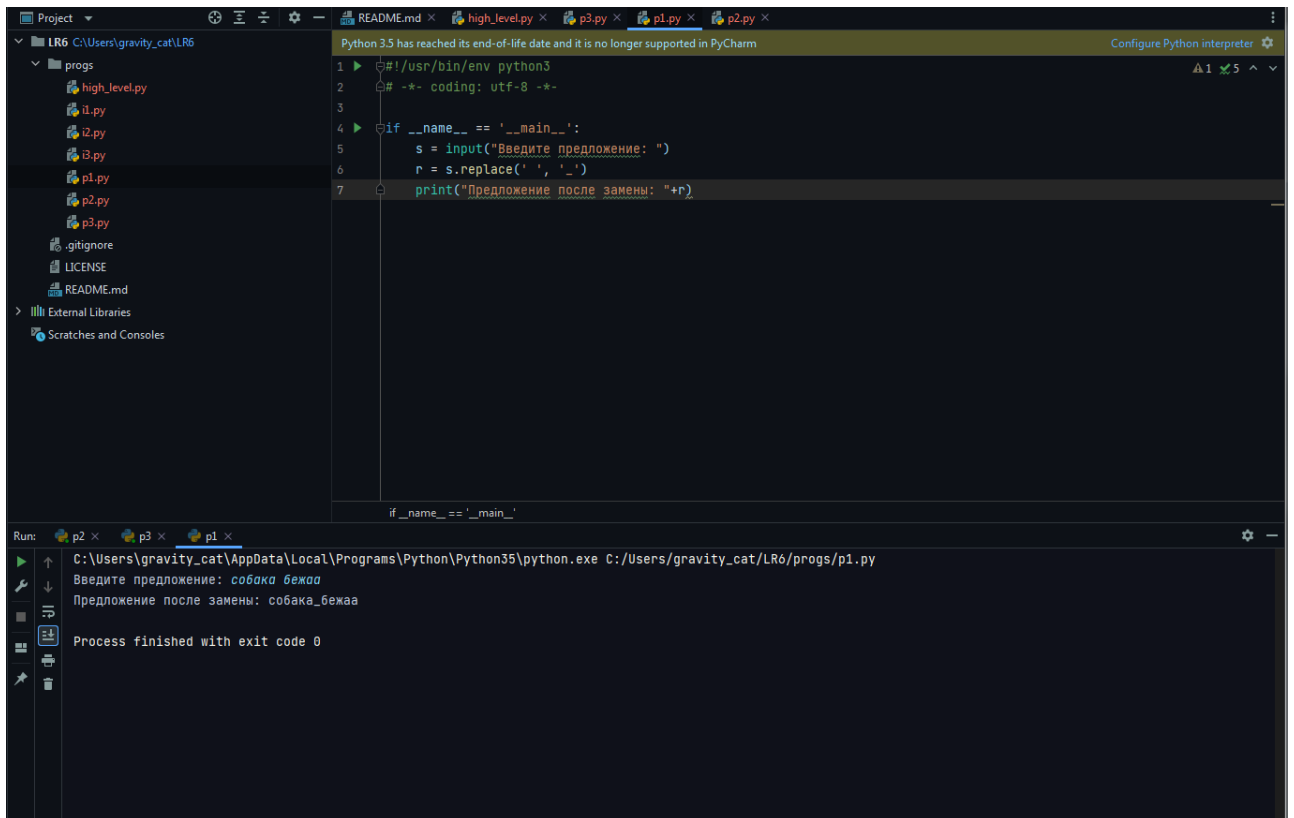


Рисунок 6.1 - Первый пример

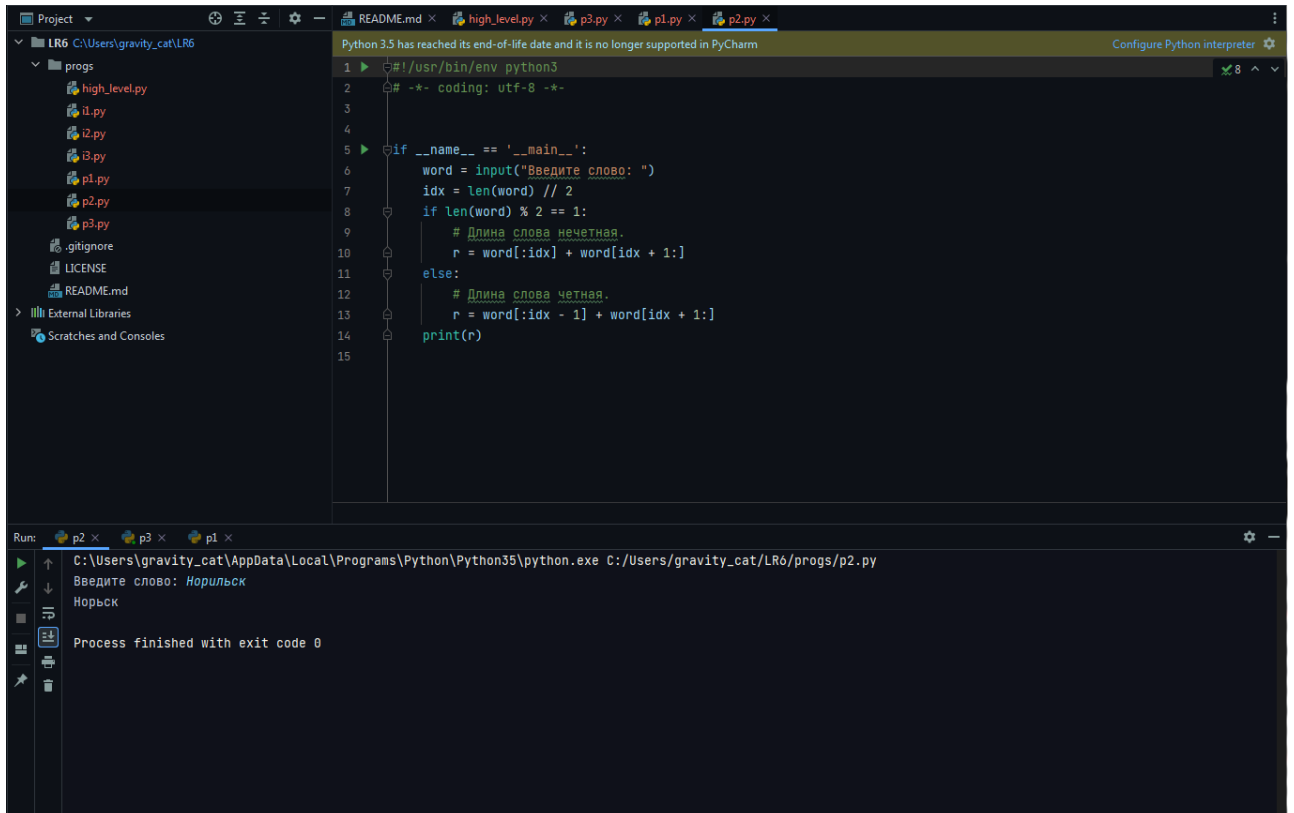


Рисунок 6.2 – Второй пример

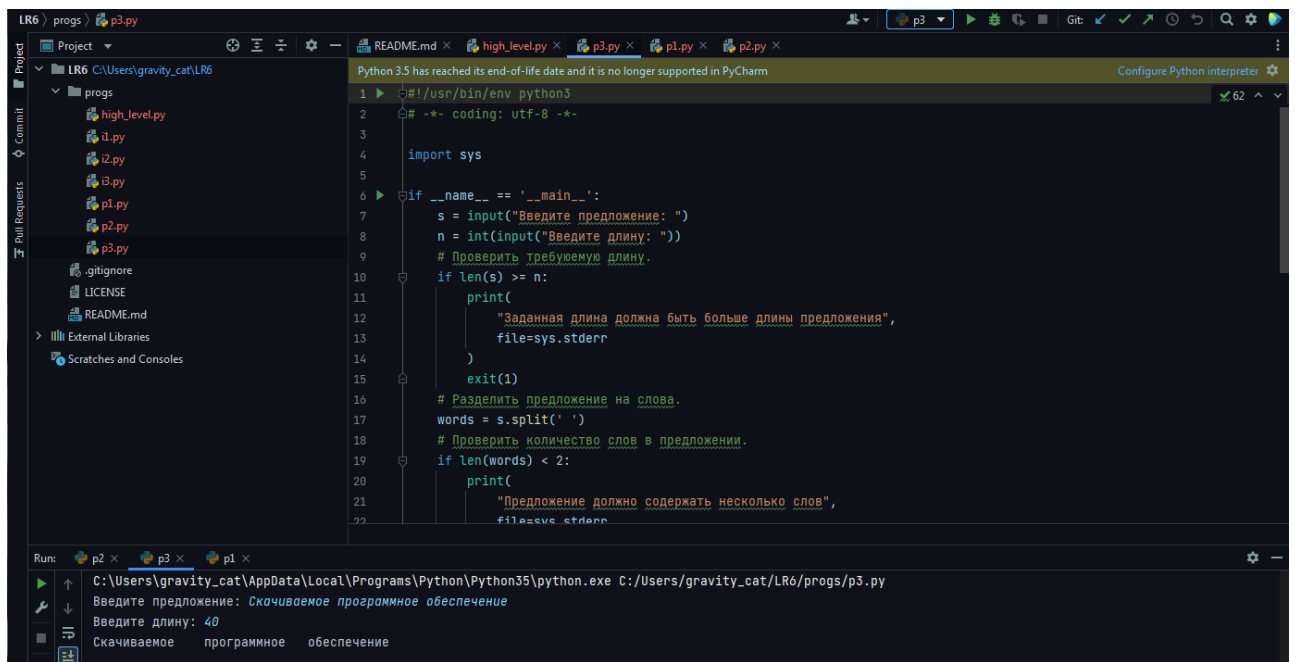


Рисунок 6.3 – Третий пример

## ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

### Вариант 5

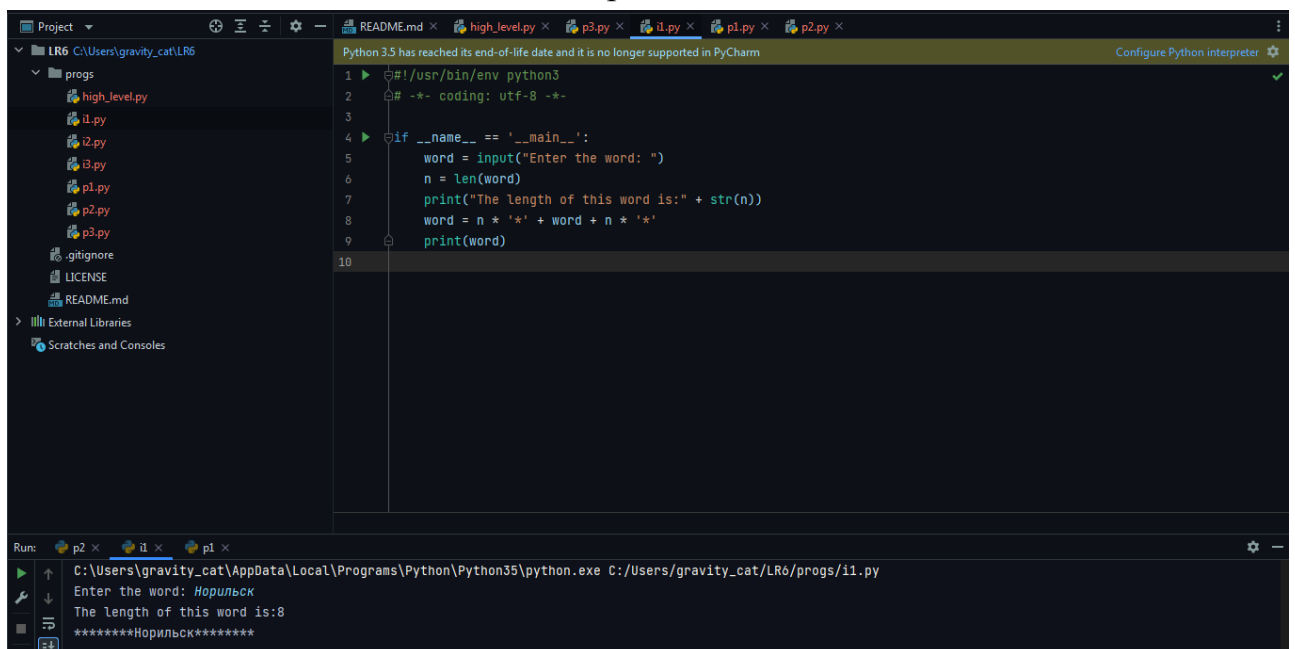


Рисунок 6.4 – Задание №1

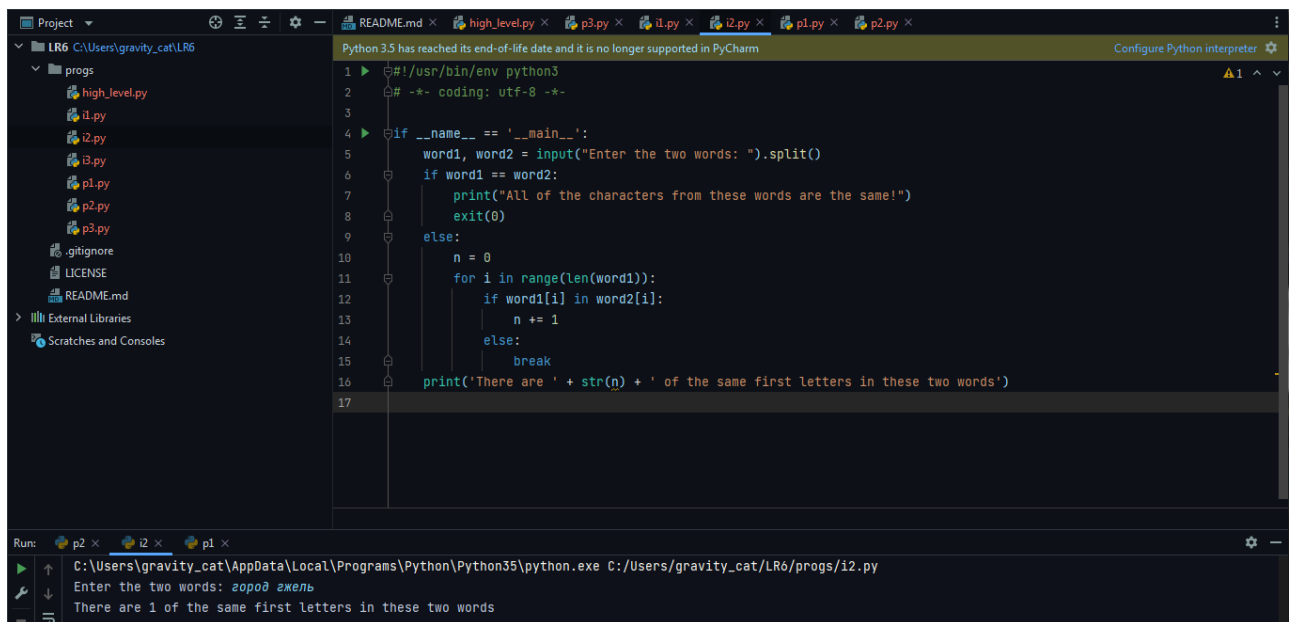


Рисунок 6.5 – Задание 2

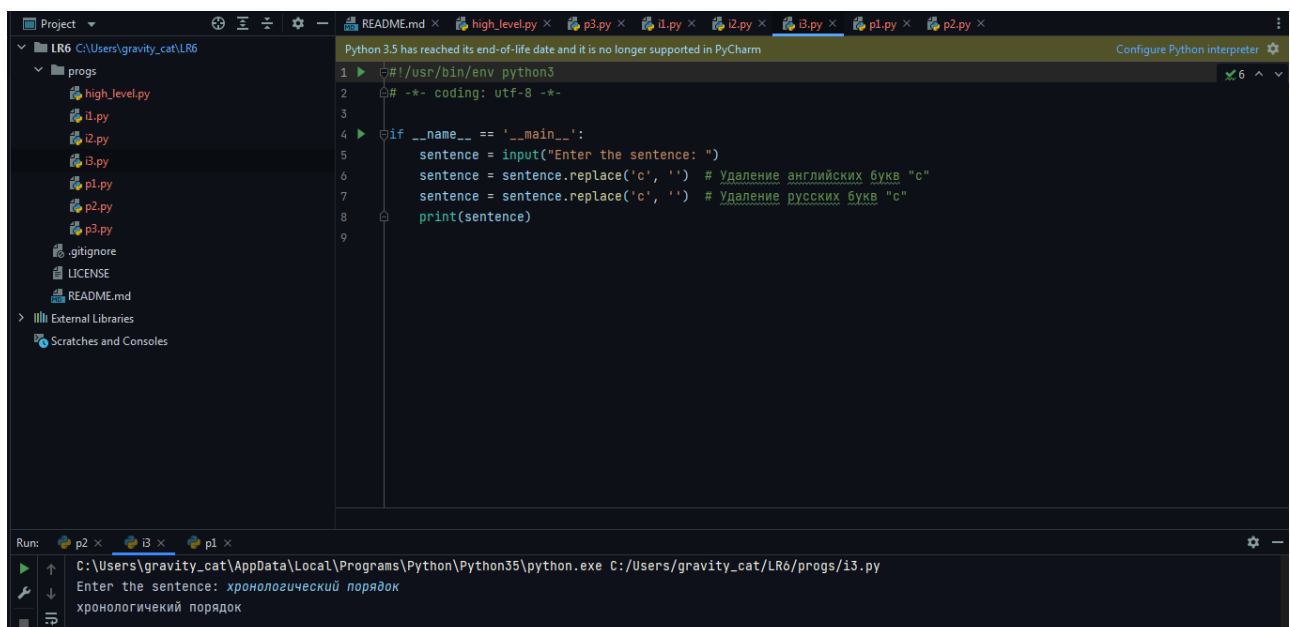


Рисунок 6.6 – Задание 3

## ЗАДАНИЕ ПОВЫШЕННОЙ СЛОЖНОСТИ

### Вариант 5

Даны два слова. Для каждой буквы первого слова (в том числе для повторяющихся в этом слове букв) определить, входит ли она во второе слово. Например, если заданные слова "информация" и "процессор", то для букв первого из них ответом должно быть: нет нет нет да да нет нет да нет нет.



Рисунок 6.7 – Задание повышенной сложности

## ОТВЕТЫ НА ВОПРОСЫ

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.tittle()`
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
32. Когда нужен индекс последнего вхождения подстроки в строку