# МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

**ФЕДЕРАЦИИ**

# Федеральное государственное автономное образовательное учреждение высшего образования

**«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

# Кафедра инфокоммуникаций Институт цифрового развития

**ОТЧЁТ**

# по лабораторной работе №2.3

Дисциплина: «Основы программной инженерии»

Тема: «Работа со строками в языке Python»

Выполнила: студентка 2 курса группы Пиж-б-о-21-1

Джолдошова Мээрим Бекболотовна

Ставрополь 2022

Цель: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python version .3.x

1. Был создан общедоступного репозиторий в GitHub в котором были добавлены gitignore, правила для работы с IDE PyCharm с ЯП Python и лицензия MIT, репозиторий был клонировал на локальный сервер и организован в соответствие с моделью ветвления git-flow.

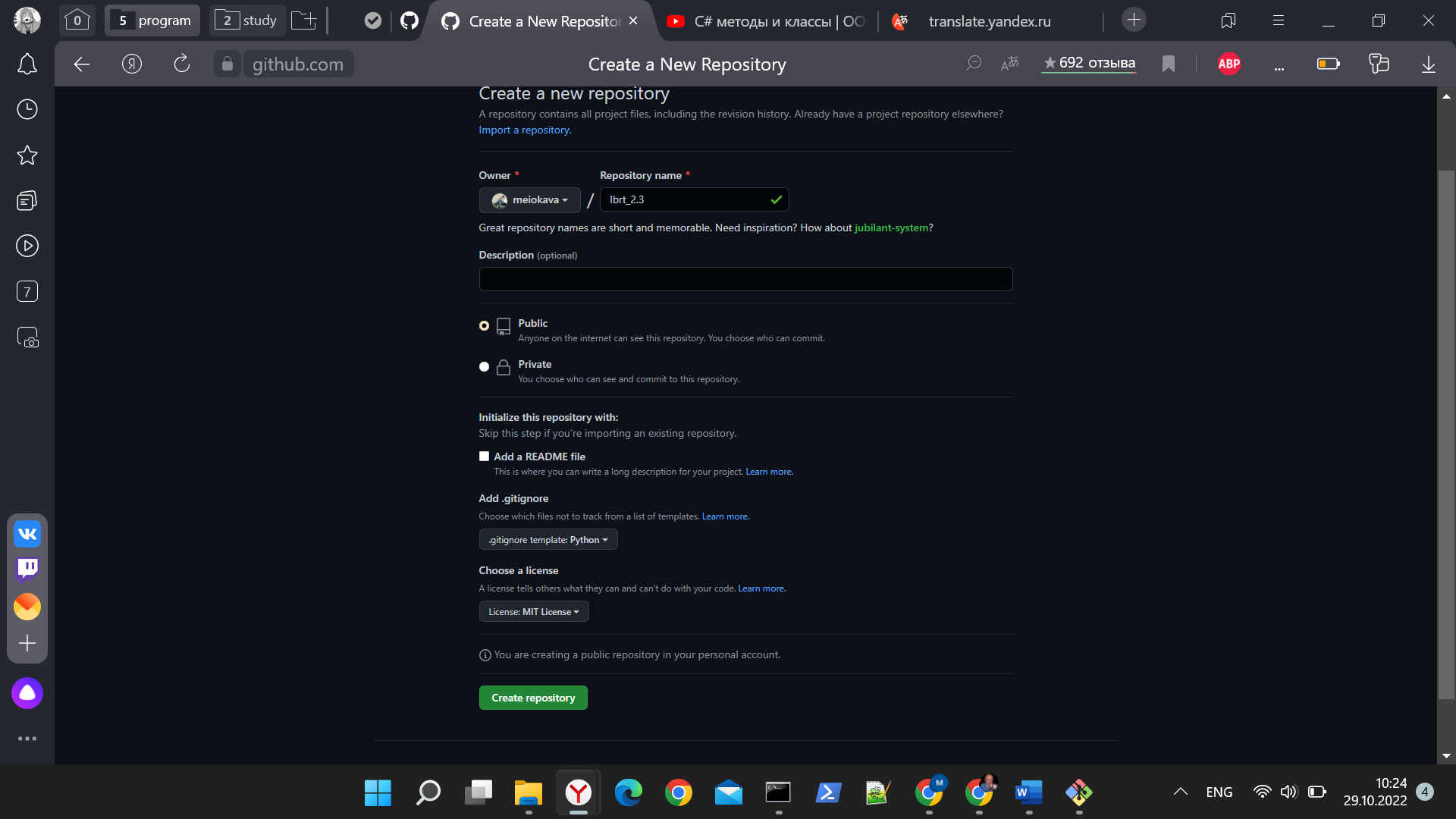


Рисунок 1.1 – Создание общедоступного репозитория

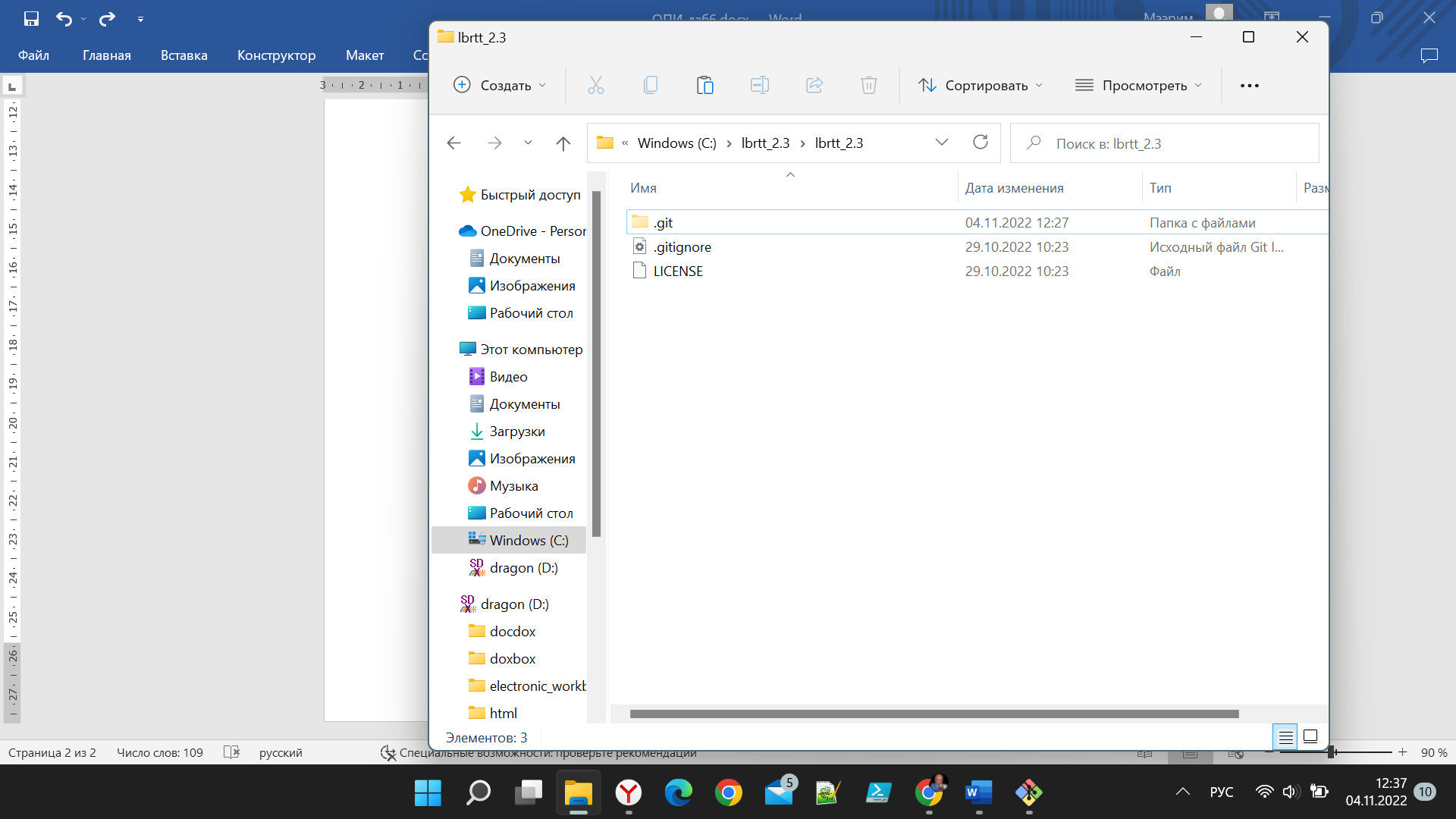


Рисунок 1.2 – Репозиторий был успешно клонирован

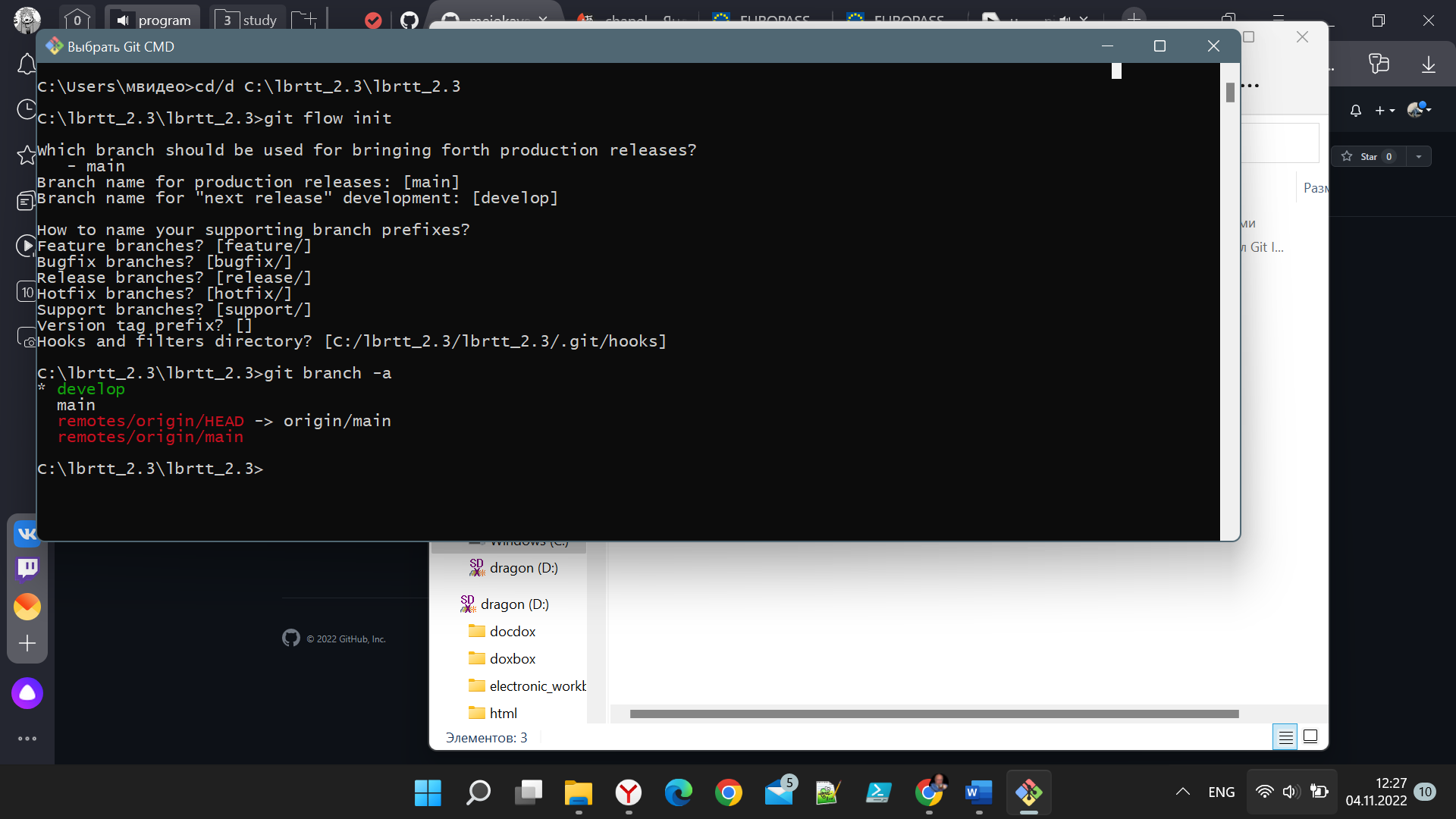


Рисунок 1.3 – Репозиторий был организован в соответствии с моделью ветвления git-flow

1. Была создана папка pycharm в которую были помещены примеры.

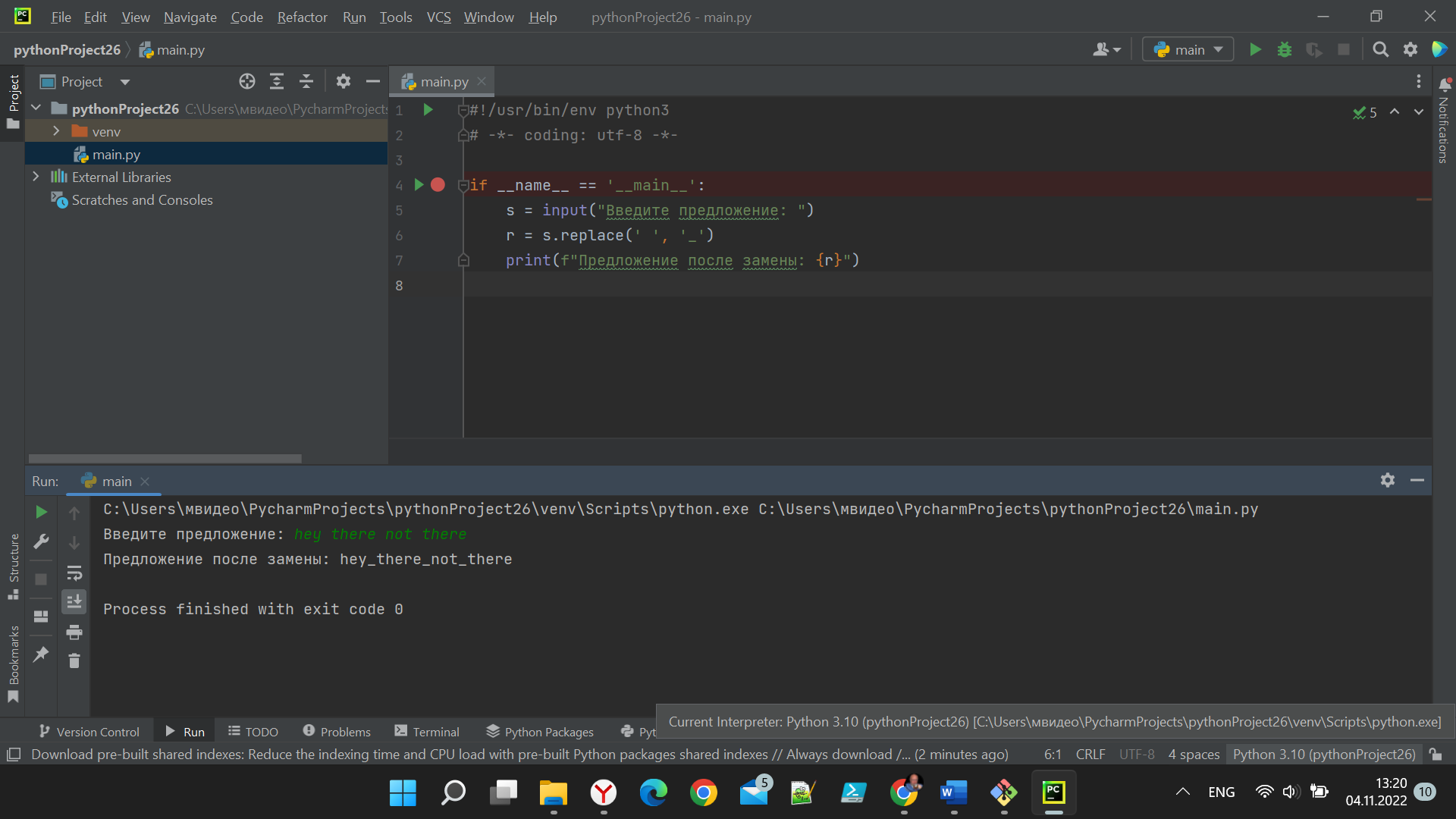


Рисунок 2.1 – Результат работы первого примера

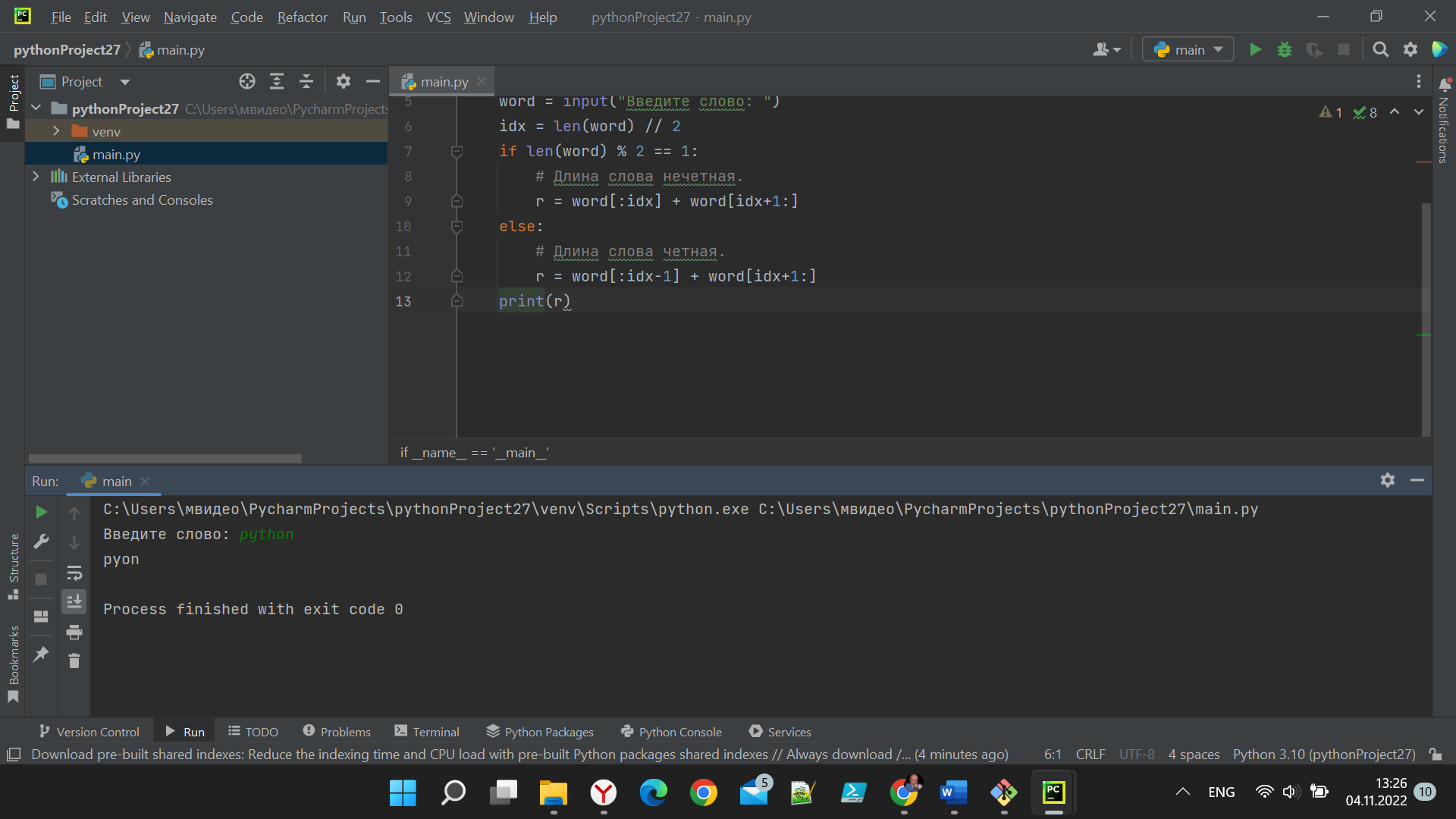


Рисунок 2.2 – Результат работы второго примера

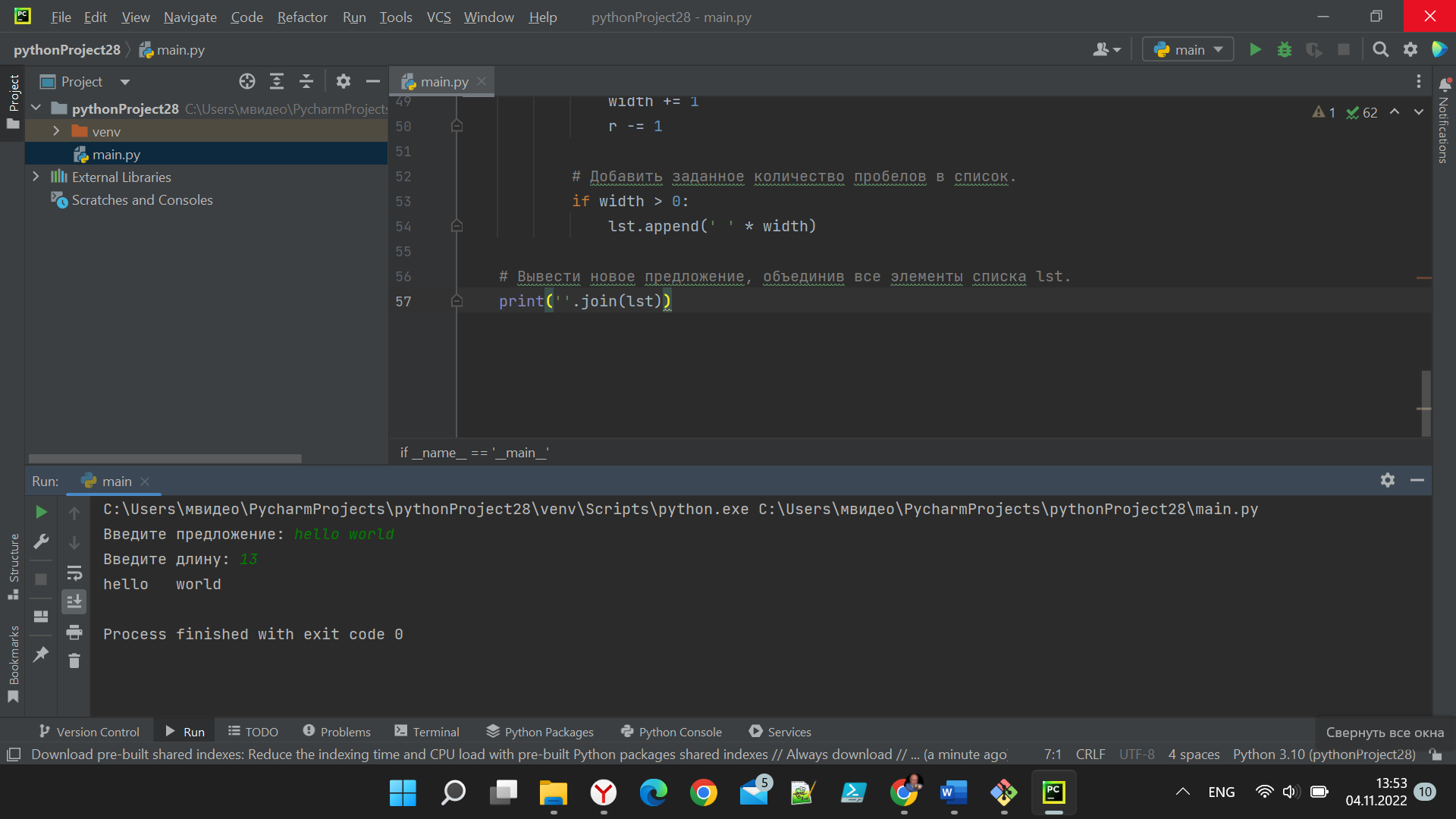


Рисунок 2.3 – Результат работы третьего примера

1. Было выполнено три индивидуальных задания в соответствии с вариантом 5

Задание 1

Дано слово. Добавить к нему в начале и конце столько звездочек, сколько букв в этом слове.

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 word = input("enter word: ")  
 print('\*' \* len(word) + word + ('\*' \* len(word)))

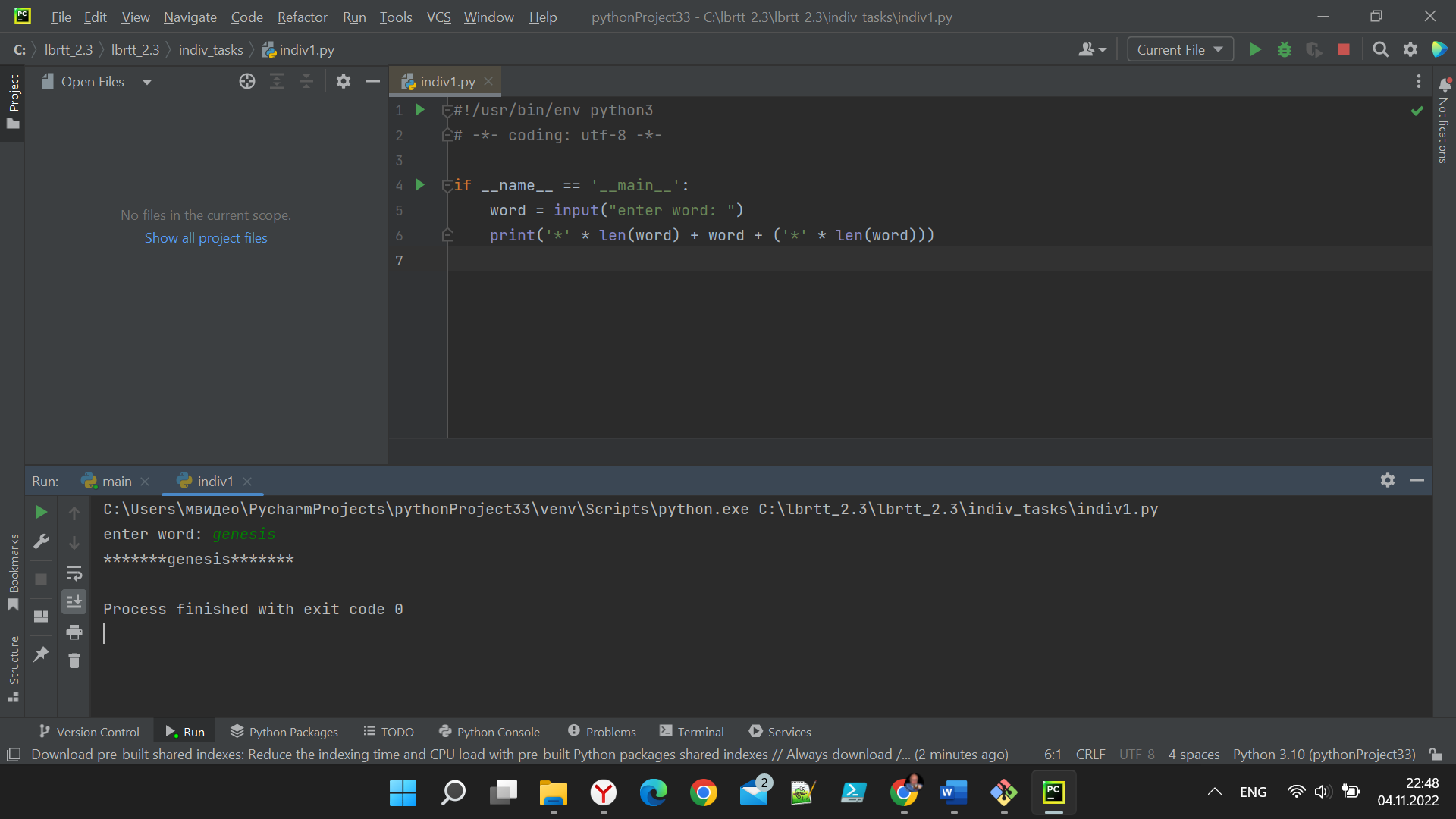


Рисунок 3.1 – Результат выполнения программы

Задание 2

Даны два слова. Определить, сколько начальных букв первого слова совпадает с начальными буквами второго слова. Рассмотреть два случая:

известно, что слова разные;

слова могут быть одинаковыми.

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 word1 = input('enter first word: ')  
 word2 = input('enter second word: ')  
 k = 0  
 for i in range(len(word1)):  
 if word1[i] == word2[i]:  
 k += 1  
 print(k)

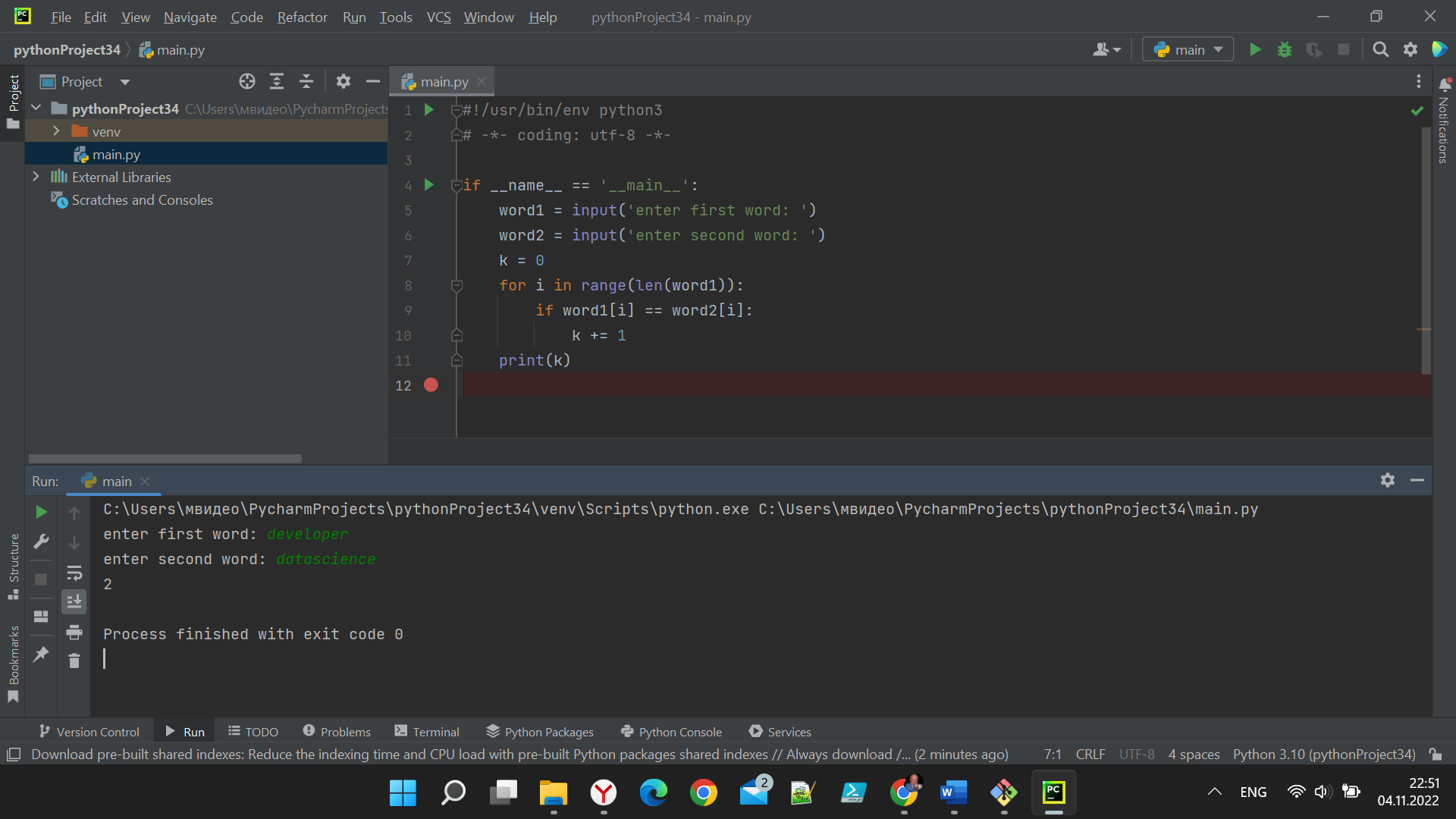


Рисунок 3.2 – Результат работы программы в первом случае

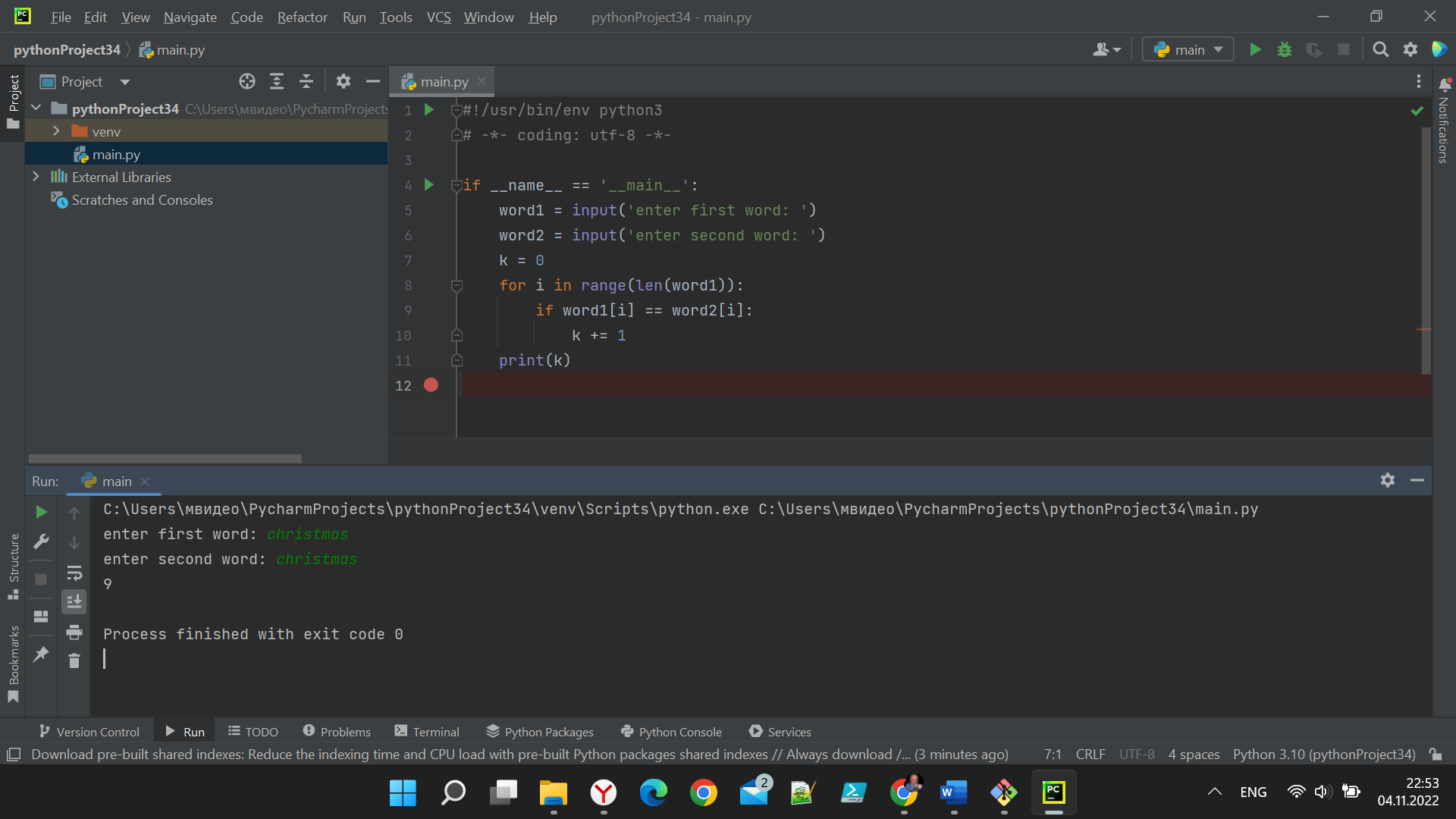


Рисунок 3.3 – Результат работы программы во втором случае

Задание 3

Дано предложение. Удалить из него все буквы с (как в кириллице так и на латинице).

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 sentence = input("enter your sentence, that will be altered: ")  
 sentence = sentence.replace('c', '') # for eng  
 sentence = sentence.replace('с', '') # for rus  
 print(sentence)

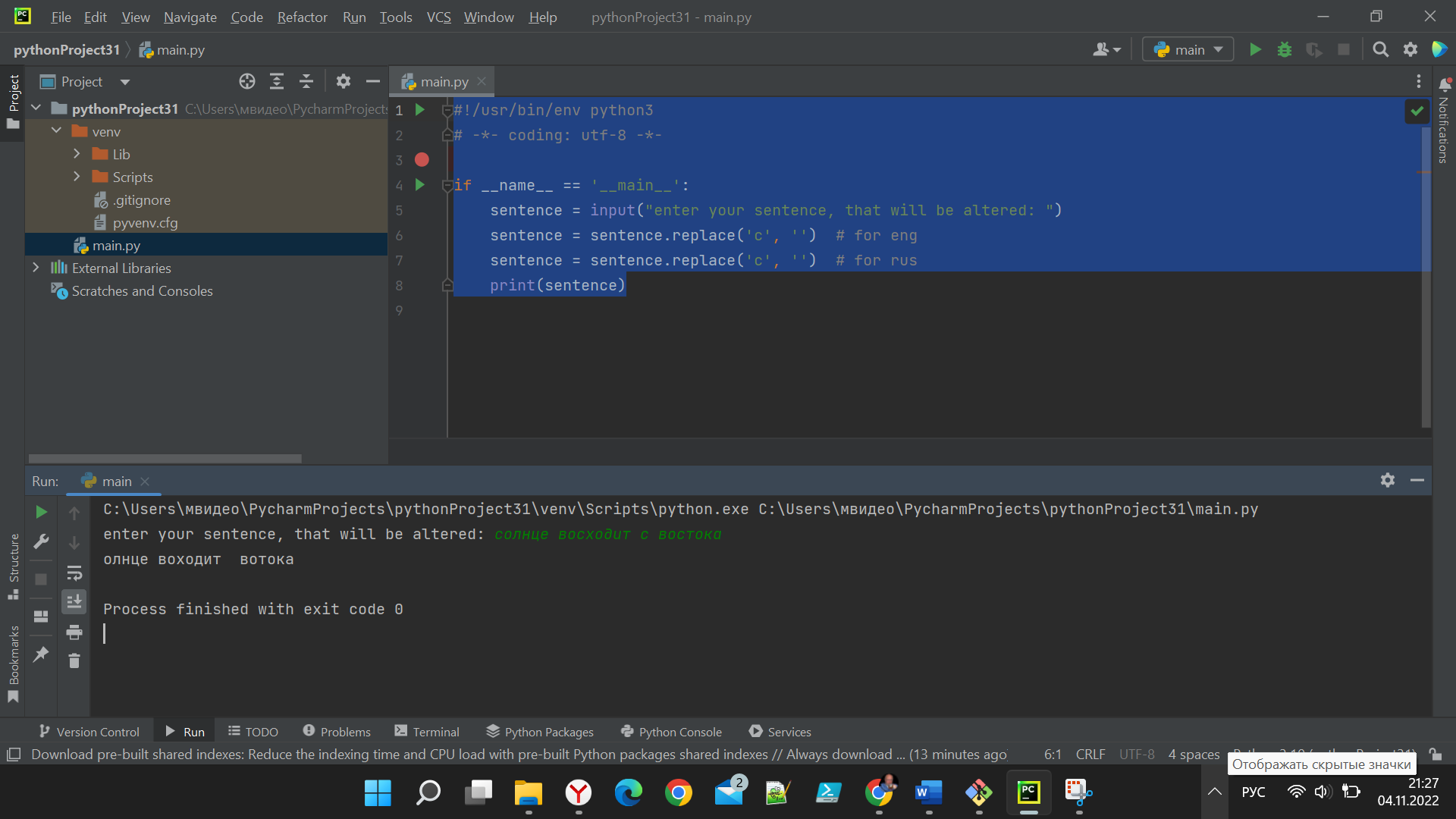


Рисунок 3.4 – Результат работы программы с предложением на русском

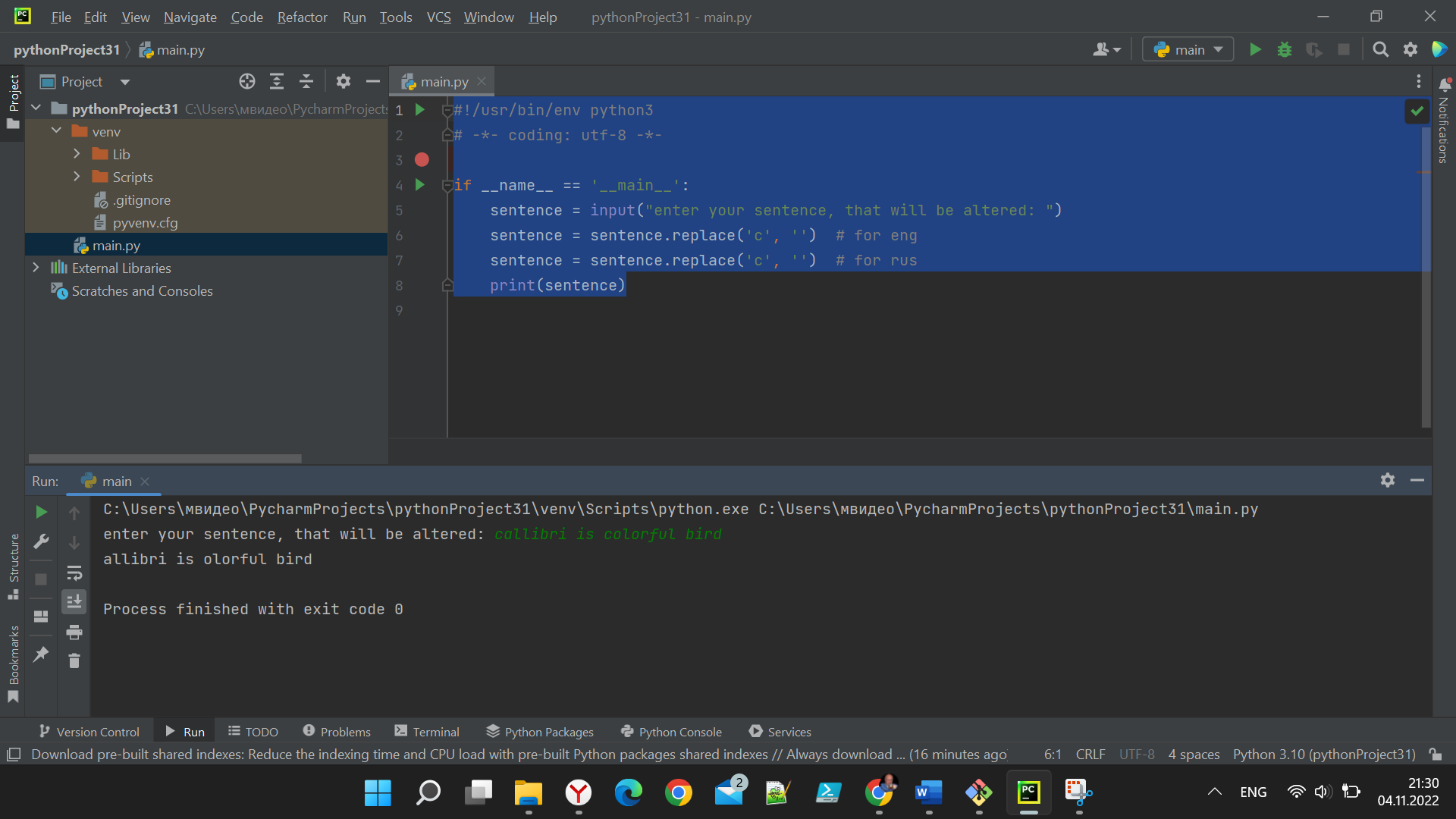


Рисунок 3.5 – Результат работы программы с предложением на английском

1. Было выполнено задание повышенной сложности согласно варианту. Были зафиксированы изменения и слита ветка develop с веткой main.

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.

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 word1 = input('enter first word: ')  
 word2 = input('enter second word: ')  
 k = 0  
 for i in word1:  
 print(k + 1, " letter ", i in word2)  
 k += 1

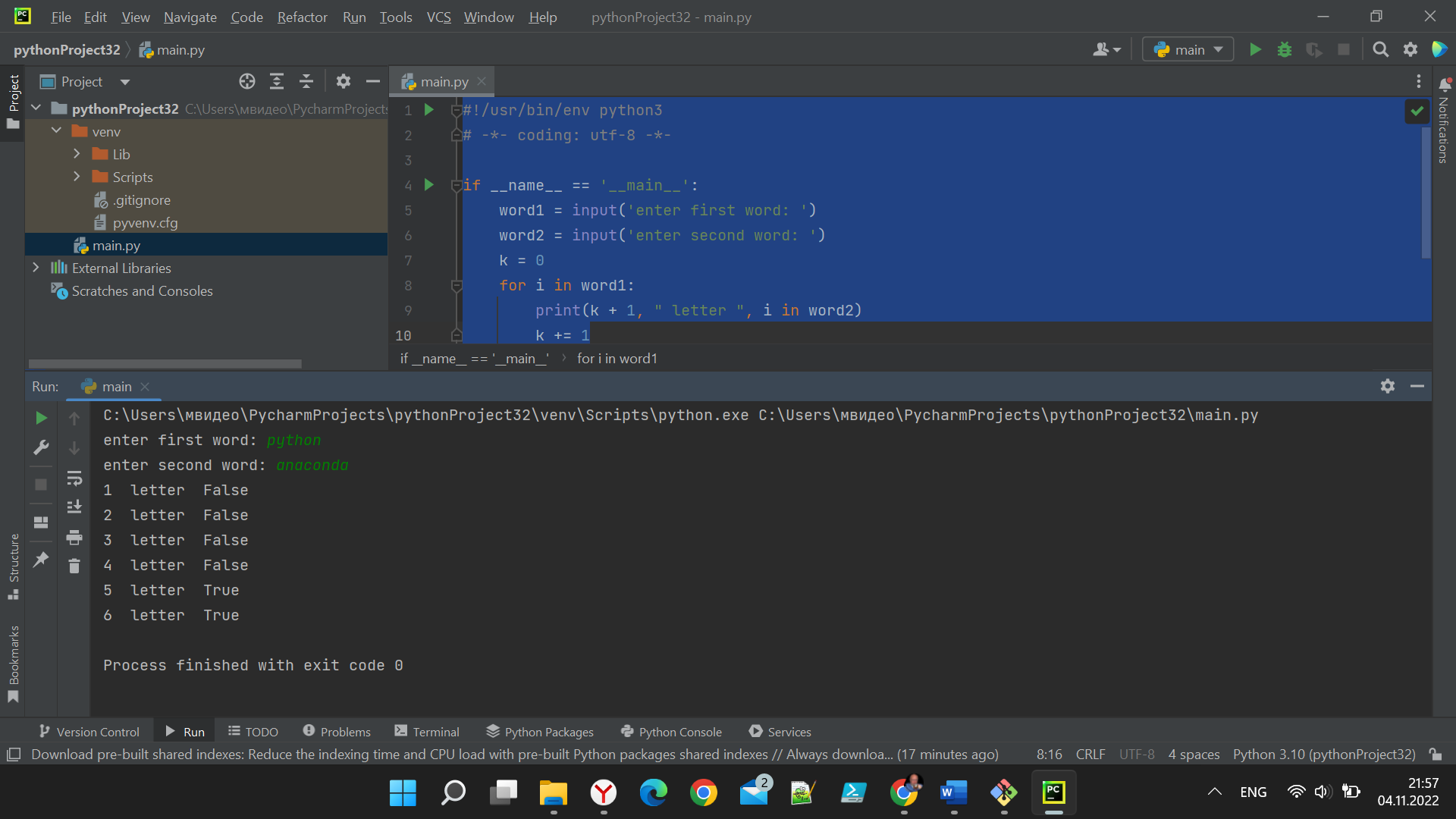


Рисунок 4.1 – Результат работы программы

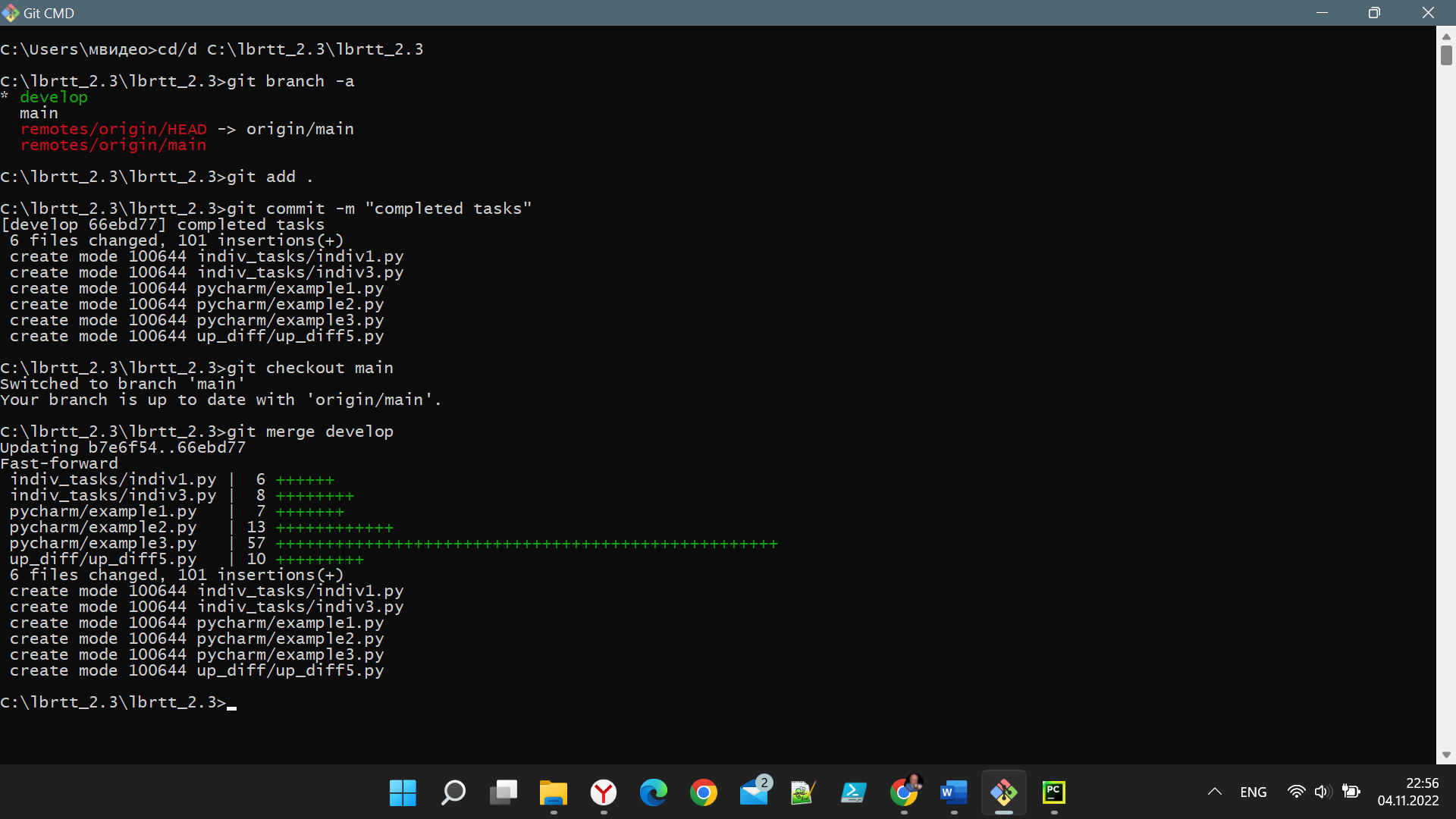


Рисунок 4.2 – Коммит изменений и слияние ветки develop с main

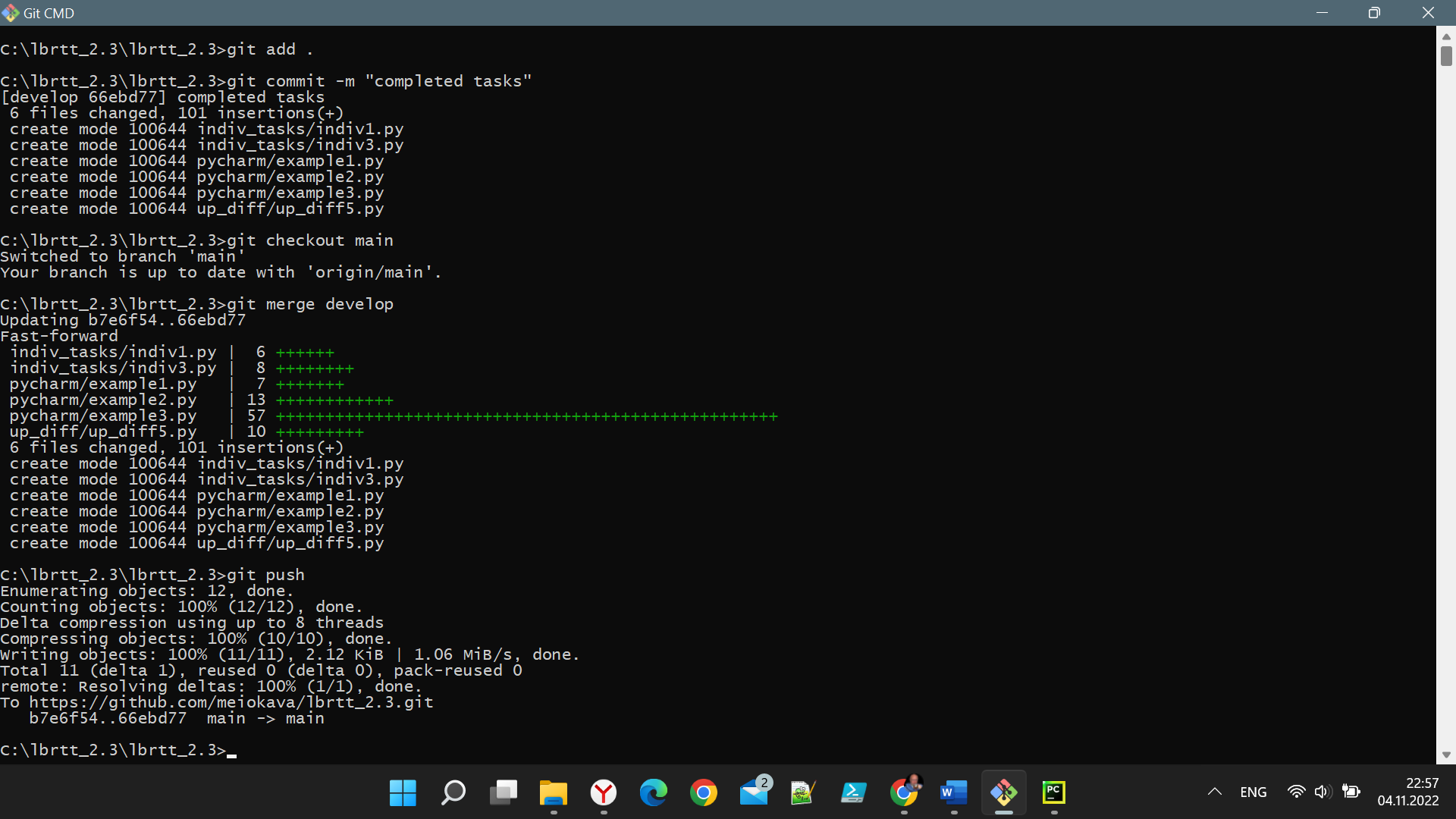


Рисунок 4.3 – Пуш на удаленный сервер

Вывод: в результате лабораторной работы были приобретены навыки по работе со строками при написании программ с помощью языка программирования Python version .3.x

Контрольные вопросы

1. Что такое строки в языке Python?

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. Какие существуют способы задания строковых литералов в языке Python?

Строки в апострофах и в кавычках, экранированные последовательности, "сырые" строки, строки в тройных апострофах или кавычках.

3. Какие операции и функции существуют для строк?

Сложение, дублирование, длина строки, длина строки, извлечение среза и т. д.

4. Как осуществляется индексирование строк?

Доступ к символам в строках основан на операции индексирования – после строки или имени переменной, ссылающейся на строку, в квадратных скобках указываются номера позиций необходимых символов.

5. Как осуществляется работа со срезами для строк?

Есть три формы срезов. Самая простая форма среза: взятие одного символа строки, а именно, S[i] — это срез, состоящий из одного символа, который имеет номер i, при этом считая, что нумерация начинается с числа 0. То есть если S = ‘Hello’, то S[0]=='H', S[1]=='e', S[2]=='l', S[3]=='l', S[4]=='o'.

Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1.

Срез с двумя параметрами: S[a:b] возвращает подстроку из b-a символов, начиная с символа c индексом a, то есть до символа с индексом b, не включая его.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. Python дает возможность изменять (заменять и перезаписывать) строки.

7. Как проверить то, что каждое слово в строке начинается с заглавной буквы?

string.istitle()

8. Как проверить строку на вхождение в неё другой строки?

string.find()

9. Как найти индекс первого вхождения подстроки в строку?

s.partition(<sep>)

10. Как подсчитать количество символов в строке?

len(s)

11. Как подсчитать то, сколько раз определённый символ встречается в строке?

s.count(<sub>)

12. Что такое f-строки и как ими пользоваться?

Эти строки улучшают читаемость кода, а также работают быстрее чем другие способы форматирования. F-строки задаются с помощью литерала «f» перед кавычками. Пример: print(f"Меня зовут {name} Мне {age} лет.")

13. Как найти подстроку в заданной части строки?

s.find(значение, начало, конец)

14. Как вставить содержимое переменной в строку, воспользовавшись методом format()?

print('{}'.format(s))

15. Как узнать о том, что в строке содержатся только цифры?

s.isdigit()

16. Как разделить строку по заданному символу букв?

str.split()

17. Как проверить строку на то, что она составлена только из строчных

s.isalpha()

18. Как проверить то, что строка начинается со строчной буквы?

s.istitle()

19. Можно ли в Python прибавить целое число к строке?

Нет

20. Как «перевернуть» строку?

s.reverse()

21. Как объединить список строк в одну строку, элементы которой разделены дефисами?

str.split(‘-’)

22. Как привести всю строку к верхнему или нижнему регистру?

s.upper() s.lower

23. Как преобразовать первый символ строки к верхнему регистру?

s.capitalize()

24. Как проверить строку на то, что она составлена только из прописных букв?

s.isupper()

25. В какой ситуации вы воспользовались бы методом splitlines() ?

s.splitlines() делит s на строки и возвращает их в списке. Любой из следующих символов или последовательностей символов считается границей строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?

s.replace(old, new)

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?

str.startswith() и str.endswith()

28. Как узнать о том, что строка включает в себя только пробелы?

s. isspace()

29. Что случится, если умножить некую строку на 3?

Asd\*3 = AsdAsdAsd

30. Как привести к верхнему регистру первый символ каждого слова в строке?

s.title()

31. Как пользоваться методом partition()?

Метод partition() разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом rfind()?

s.rfind(<sub>) возвращает индекс последнего вхождения подстроки

<sub> в s , который соответствует началу <sub>.