

Северо-Кавказский федеральный университет
Кафедра инфокоммуникаций СКФУ

Отчет
По лабораторной работе №4
По предмету: «Основы
кроссплатформенного программирования»

Исполнитель:

Студента группы ИТС-б-з-22-1

Направление подготовки 11.03.02

Инфокоммуникационные

технологии и системы связи

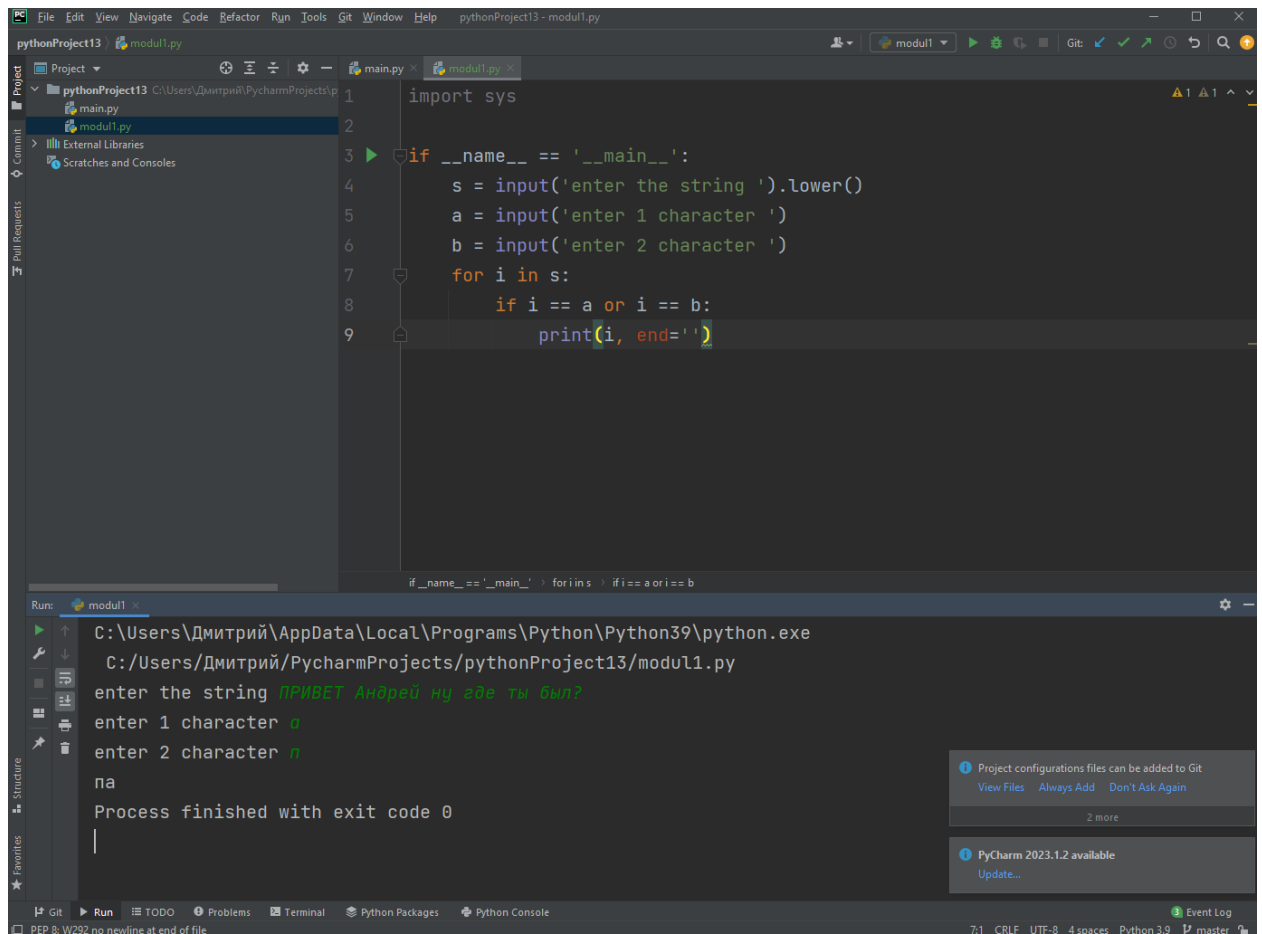
Пальников Станислав Петрович
(Ф.И.О.)

Руководитель дисциплины:

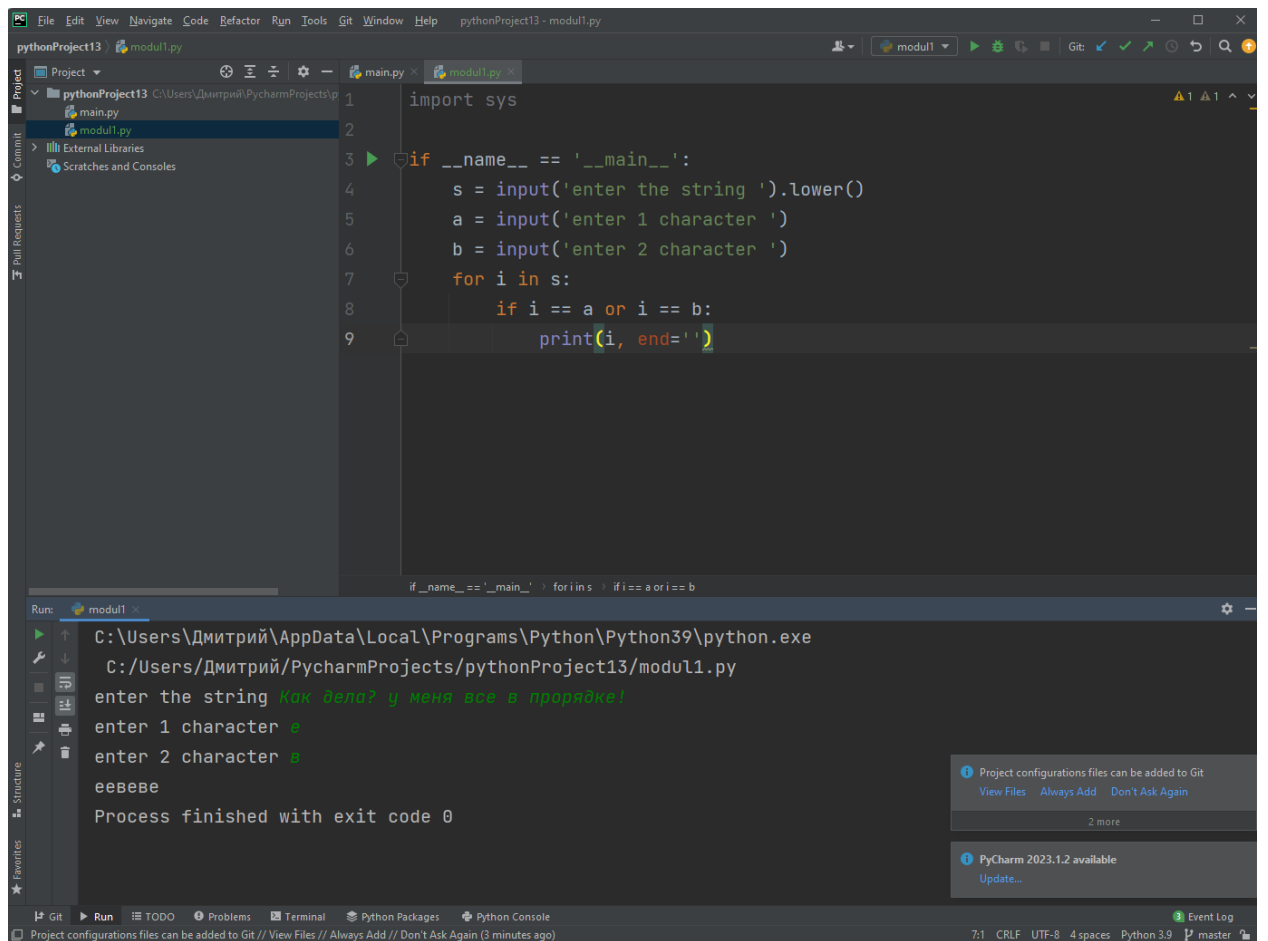
Воронкин Роман Александрович

Ставрополь, 2023

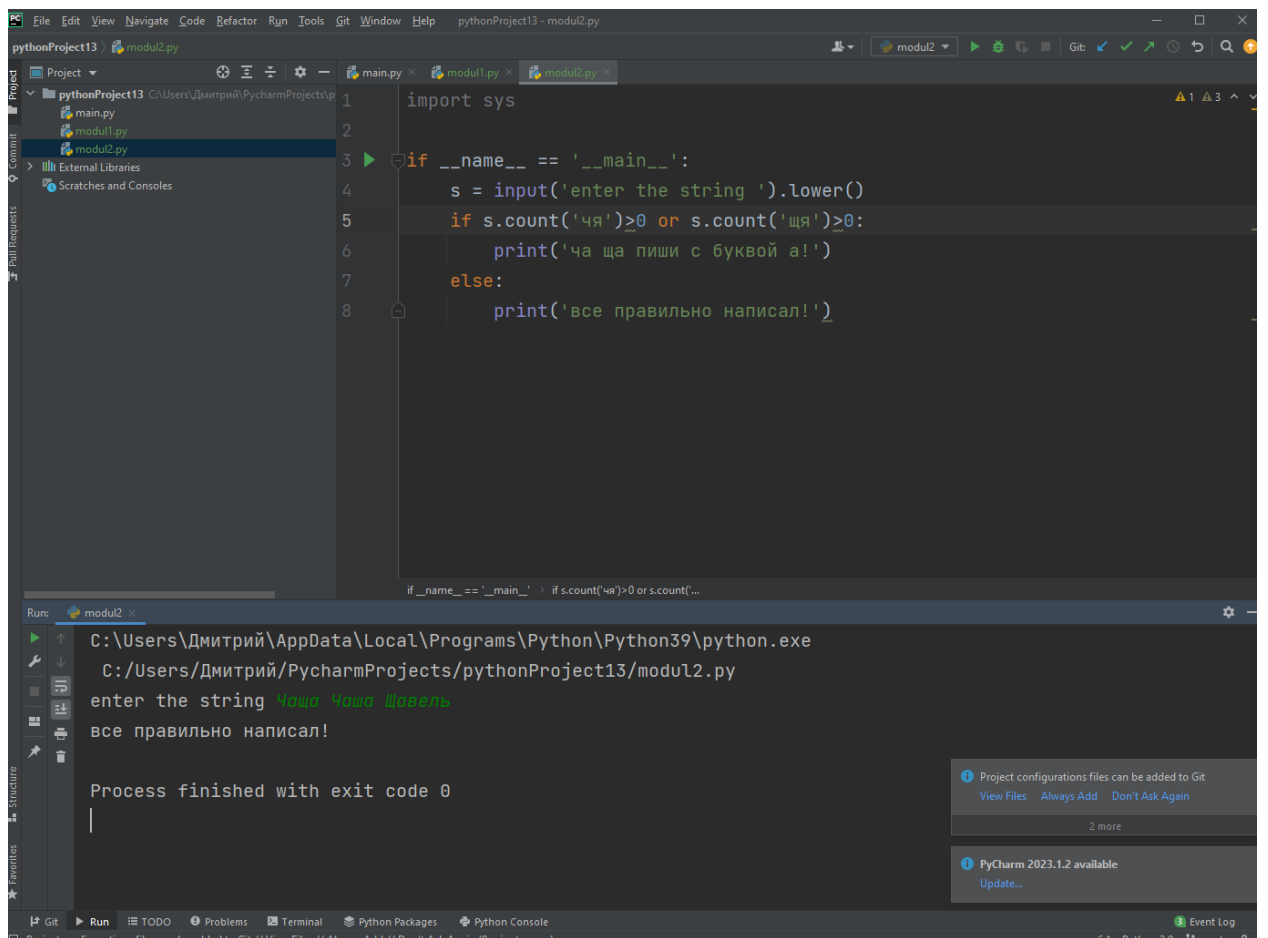
Ход выполнения работы:



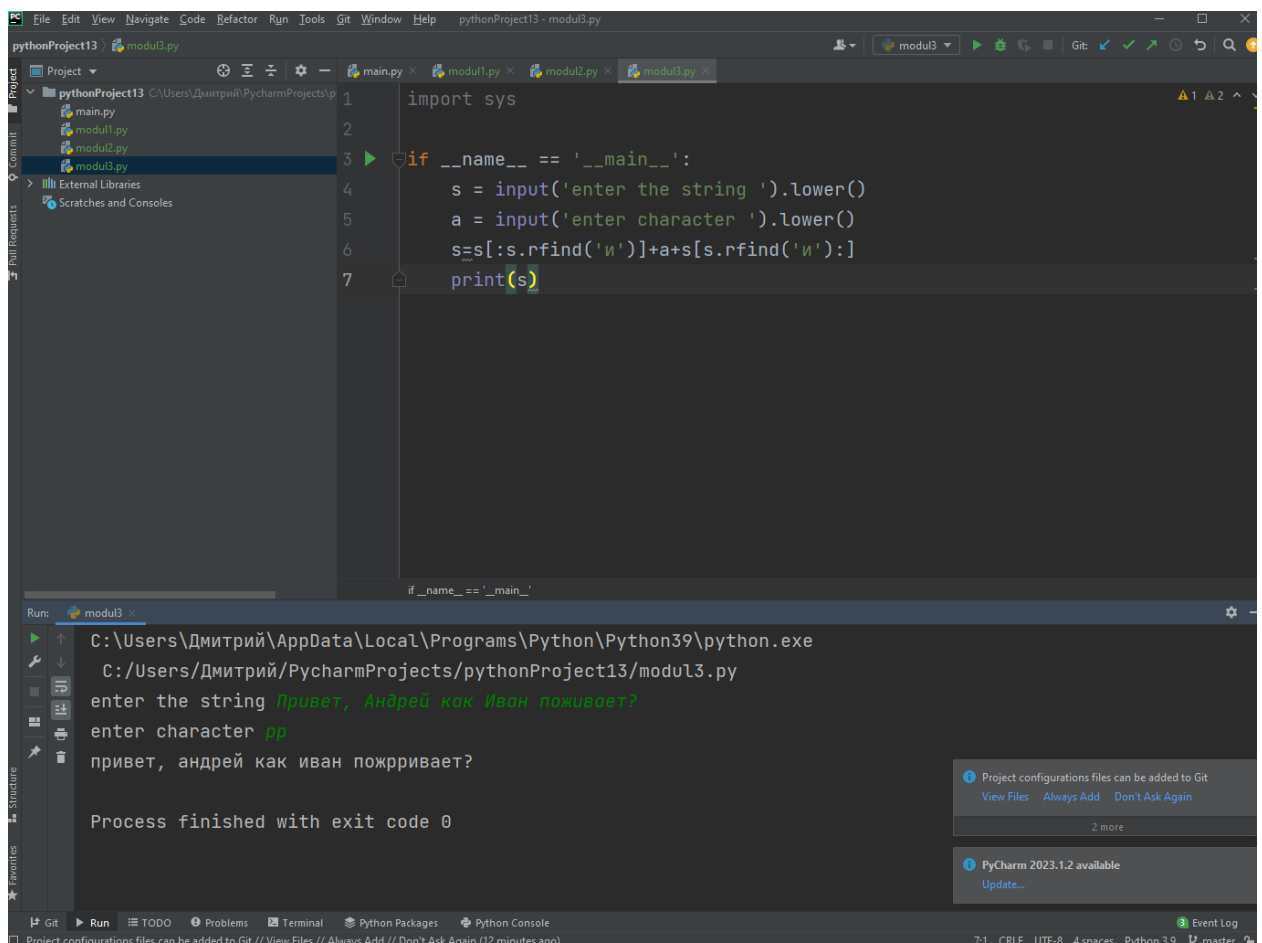
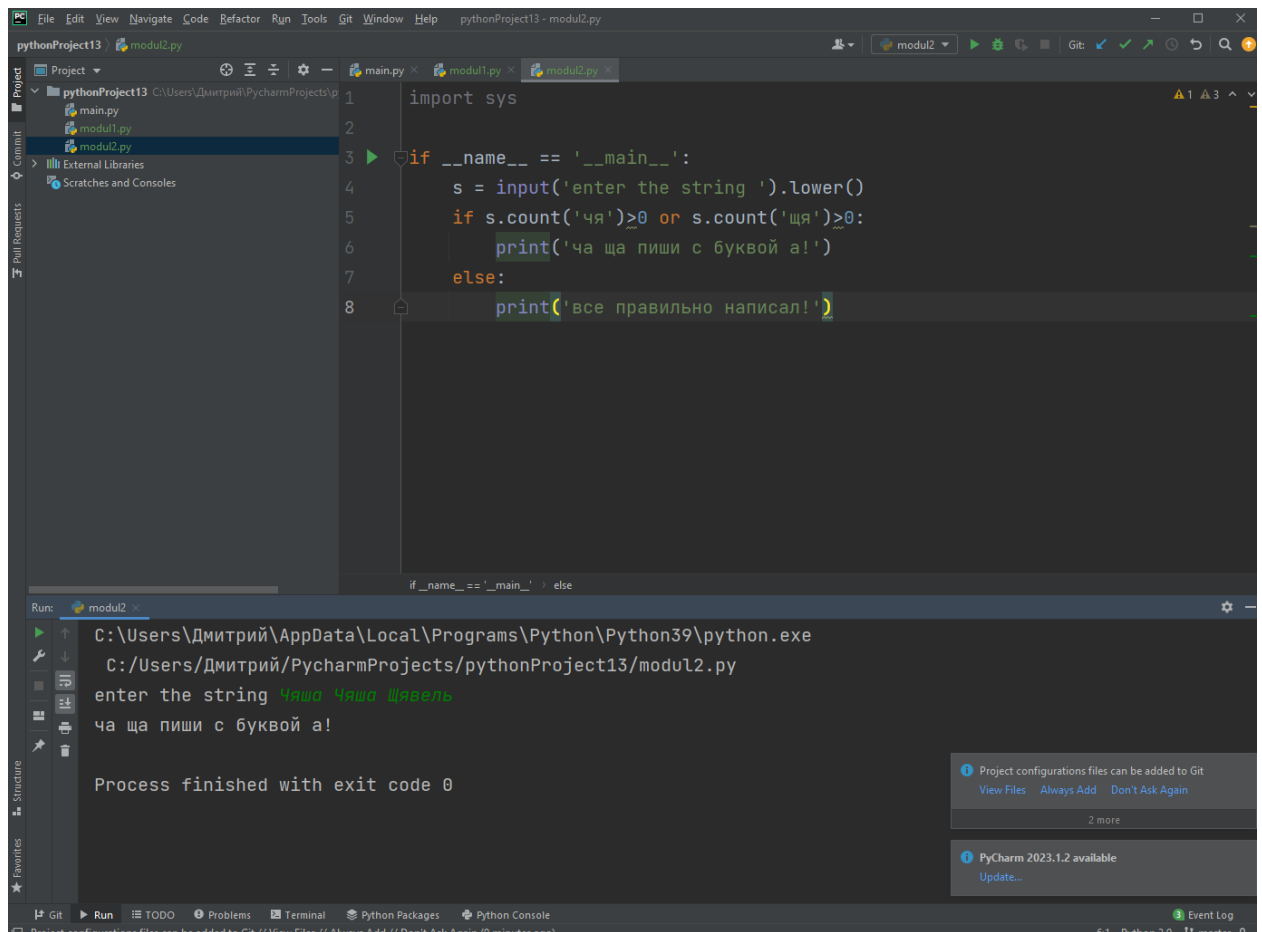
```
File Edit View Navigate Code Refactor Run Tools Git Window Help pythonProject13 - modul1.py
pythonProject13 modul1.py
Project
  pythonProject13 C:\Users\Дмитрий\PycharmProjects\p
    main.py
    modul1.py
  External Libraries
  Scratches and Consoles
Run
  modul1
  C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe
  C:/Users/Дмитрий/PycharmProjects/pythonProject13/modul1.py
  enter the string ПРИВЕТ Андрей ну где ты был?
  enter 1 character а
  enter 2 character н
  па
  Process finished with exit code 0
  Project configurations files can be added to Git
  View Files Always Add Don't Ask Again
  2 more
  PyCharm 2023.1.2 available
  Update...
  Git Run TODO Problems Terminal Python Packages Python Console
  BFP 3: W222 no newline at end of file 71 CRLF UTF-8 4 spaces Python 3.9 master
```

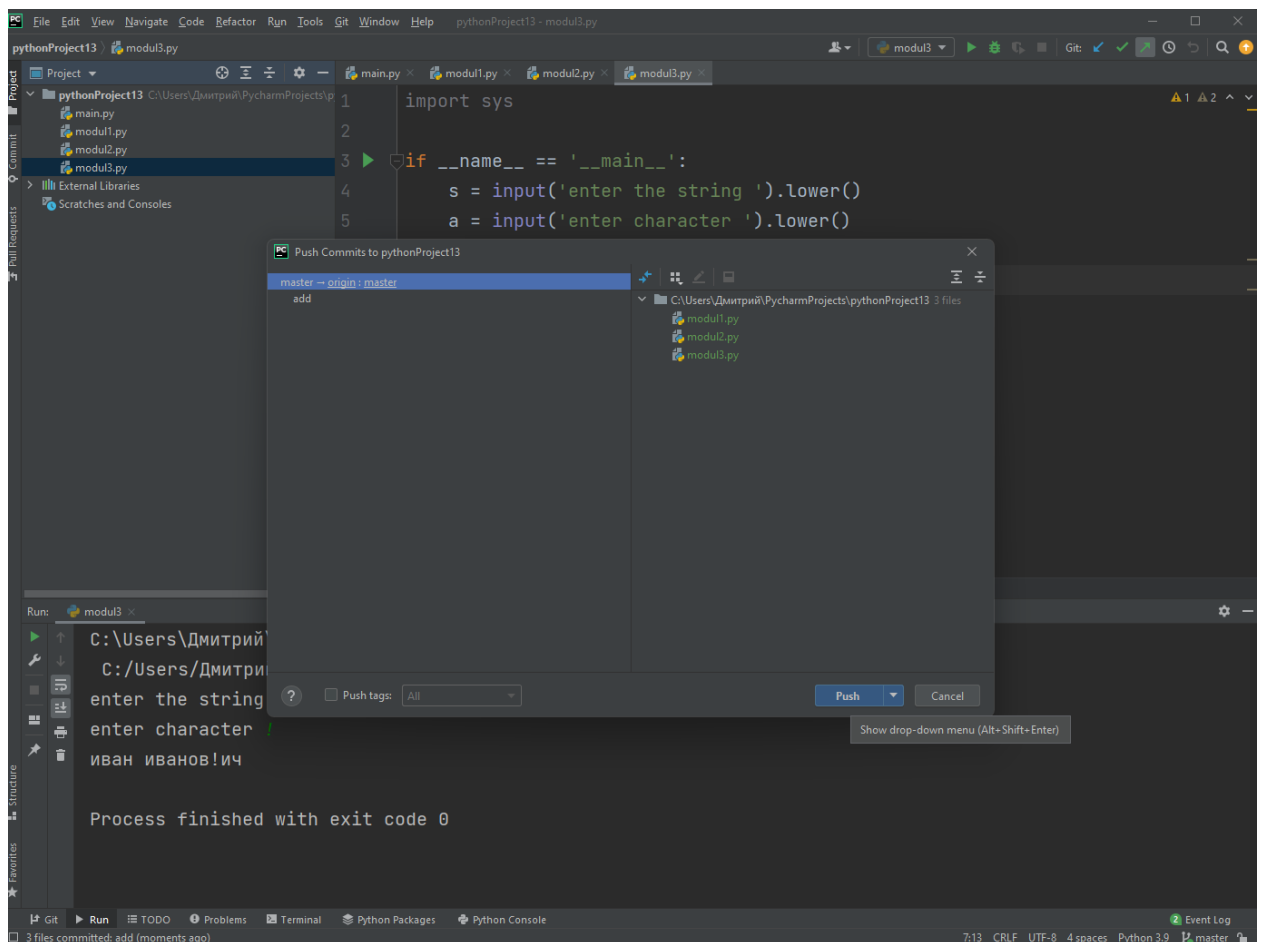
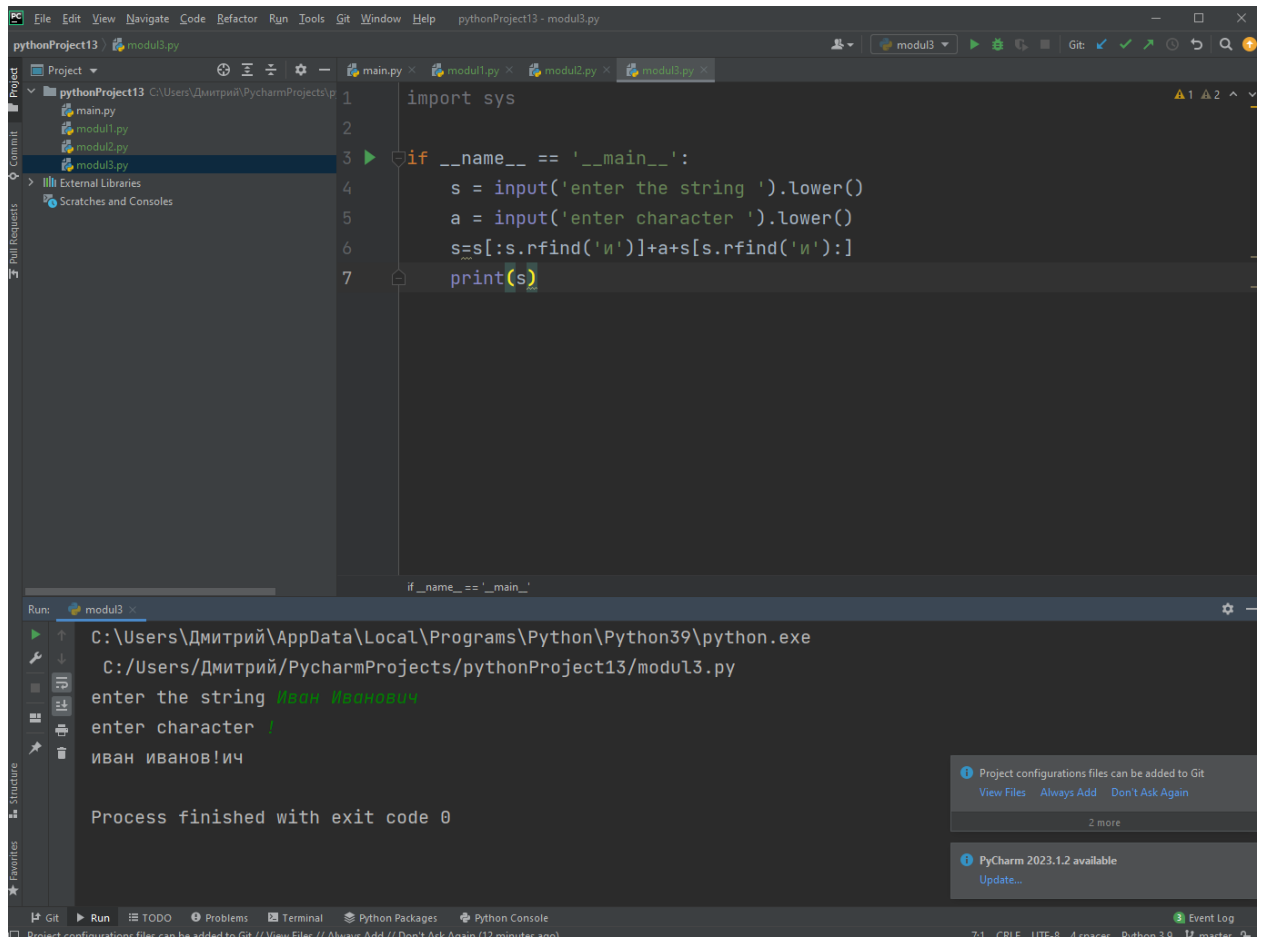


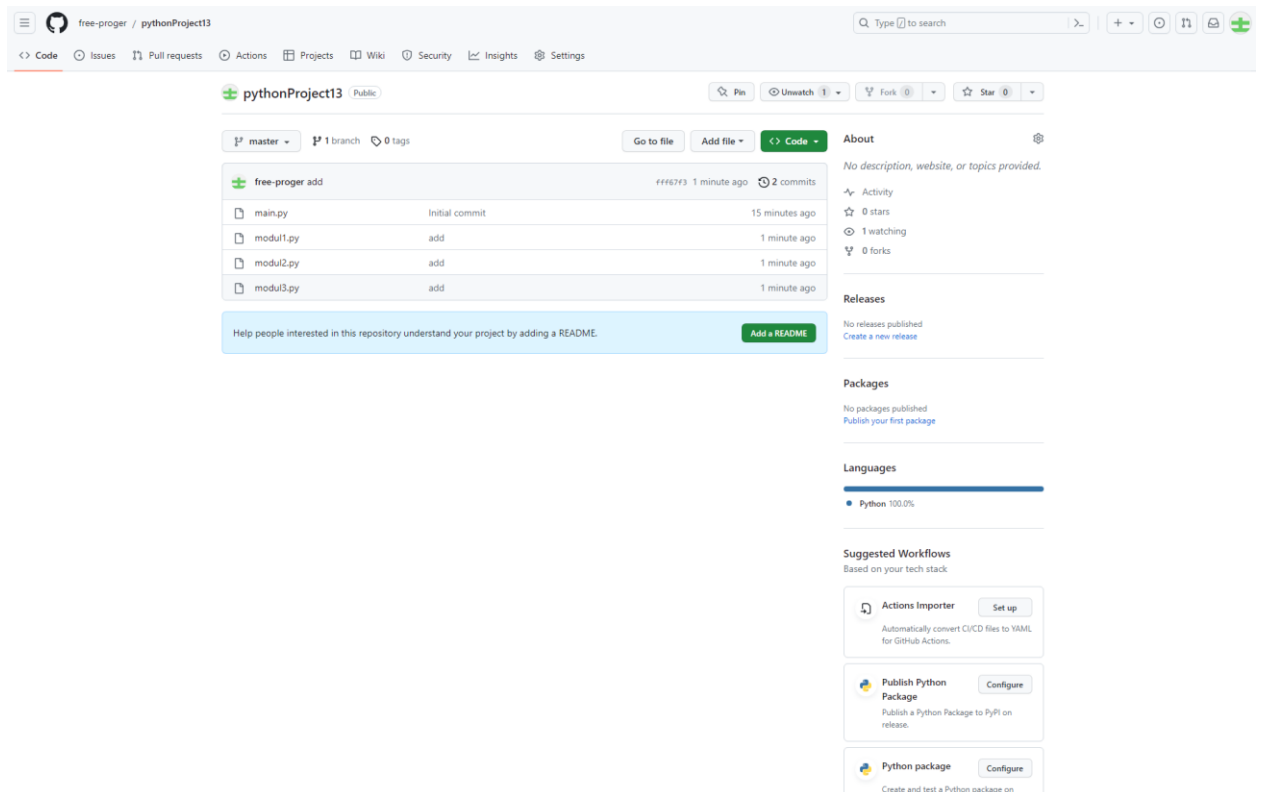
```
File Edit View Navigate Code Refactor Run Tools Git Window Help pythonProject13 - modul1.py
pythonProject13 C:\Users\Дмитрий\PycharmProjects\pythonProject13
main.py modul1.py
1 import sys
2
3 if __name__ == '__main__':
4     s = input('enter the string ').lower()
5     a = input('enter 1 character ')
6     b = input('enter 2 character ')
7     for i in s:
8         if i == a or i == b:
9             print(i, end='')
Run: modul1
C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe
C:/Users/Дмитрий/PycharmProjects/pythonProject13/modul1.py
enter the string Как дела? у меня все в порядке!
enter 1 character е
enter 2 character е
ееееее
Process finished with exit code 0
Project configurations files can be added to Git
View Files Always Add Don't Ask Again
2 more
PyCharm 2023.1.2 available
Update...
```



```
File Edit View Navigate Code Refactor Run Tools Git Window Help pythonProject13 - modul2.py
pythonProject13 C:\Users\Дмитрий\PycharmProjects\pythonProject13
main.py modul1.py modul2.py
1 import sys
2
3 if __name__ == '__main__':
4     s = input('enter the string ').lower()
5     if s.count('ч')>0 or s.count('щ')>0:
6         print('ча ща пиши с буквой а!')
7     else:
8         print('все правильно написал!')
Run: modul2
C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe
C:/Users/Дмитрий/PycharmProjects/pythonProject13/modul2.py
enter the string Чаша Чаша Щавель
все правильно написал!
Process finished with exit code 0
Project configurations files can be added to Git
View Files Always Add Don't Ask Again
2 more
PyCharm 2023.1.2 available
Update...
```







Ответы на 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 - служебные символы,

"Сырые" строки - подавляют экранирование, Строки в тройных апострофах или кавычках, Строковые операторы, Оператор сложения строк +, Оператор умножения строк *, Оператор принадлежности подстроки in.

3. Какие операции и функции существуют для строк?

Функция `ord(c)` возвращает числовое значение для заданного символа

```
. >>> ord('a')
```

```
>>> ord('#')
```

```
35
```

Функция `ord()` также возвращает числовые значения для символов Юникода:

```
>>> ord('€')
```

```
8364
```

```
>>> ord('Σ')
```

```
8721
```

Функция `chr(n)` возвращает символьное значение для данного целого числа. `chr()` действует обратно `ord()` . Если задано числовое значение `n` , `chr(n)` возвращает строку, представляющую символ `n` :

```
>>> chr(97)
```

```
'a'
```

```
>>> chr(35)
```

```
'#'
```

`chr()` также обрабатывает символы Юникода:

```
>>> chr(8364)
```

```
'€'
```

```
>>> chr(8721)
```

```
'Σ'
```

Функция `len(s)` возвращает длину строки. `len(s)` возвращает количество символов в строке `s` :

```
>>> s = 'Простая строка.'
```

```
>>> len(s)
```

```
15
```

Функция `str(obj)` возвращает строковое представление объекта. Практически любой объект в Python может быть представлен как строка. `str(obj)` возвращает строковое представление объекта `obj` :

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

4. Как осуществляется индексирование строк?

Индексация строк начинается с нуля: у первого символа индекс 0 , следующего 1 и так далее. Индекс последнего символа в python — “длина строки минус один”.

Отдельные символы доступны по индексу следующим образом:

```
>>> s = 'foobar'
>>> s[0]
'f'
>>> s[1]
'o'
>>> s[3]
'b'
>>> s[5]
'r'
```

5. Как осуществляется работа со срезами для строк?

Python также допускает возможность извлечения подстроки из строки, известную как “string slice”. Если s это строка, выражение формы s[m:n] возвращает часть s , начинающуюся с позиции m , и до позиции n , но не включая позицию:


```
>>> s = 'python'
```

```
>>> s[2:5]
```

```
'tho'
```

6. Почему строки Python относятся к неизменяемому типу данных?

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

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

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

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

Python также предоставляет оператор принадлежности, который можно использоваться для манипуляций со строками. Оператор `in` возвращает `True` , если подстрока входит в строку, и `False` , если нет.

9. Как найти индекс первого вхождения подстроки в строку?

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”.

10. Как подсчитать количество символов в строке?

`len(s)` возвращает количество символов в строке `s`.

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

Использование `count()` — самый обычный метод в Python, для того чтобы получить вхождения любого элемента в любом контейнере.

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

Одной простой особенностью f-строк, которые вы можете начать использовать сразу, является интерполяция переменной. Вы можете указать

имя переменной непосредственно в f-строковом литерале (f'string'), и python заменит имя соответствующим значением.

13. Как найти подстроку в заданной части строки?

`string.index(<sub>[,<start>[,<end>]])` ищет в строке заданную подстроку.

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

Если для подстановки требуется только один аргумент, то значение - сам аргумент:

Пример:

```
>>> 'Hello, {}!'.format('Vasya')
```

```
'Hello, Vasya!'
```

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

`string.isdigit()` определяет, состоит ли строка из цифр

16. Как разделить строку по заданному символу?

метод `split()`, который разбивает строку по заданному символу или по нескольким символам.

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

`s.islower()` возвращает `True`, если строка `s` не пустая, и все содержащиеся в нем буквенные символы строчные, а `False` если нет.

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

метод `islower()` для первого символа строки.

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

Можно если преобразовать целое число в строку

20. Как «перевернуть» строку?

Перевернуть строку можно с помощью среза `"[::-1]"`

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

Метод `join()` умеет объединять элементы списков в строки, разделяя отдельные строки с использованием заданного символа.

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

Для решения этих задач можно воспользоваться методами `upper()` и `lower()`, которые, соответственно, приводят все символы строк к верхнему и нижнему регистрам.

23. Как преобразовать первый и последний символы строки к верхнему регистру?

`string.swapcase()` меняет регистр буквенных символов на противоположный.

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

`string.isalpha()` определяет, состоит ли строка только из букв.

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

`s.splitlines()` делит `s` на строки и возвращает их в списке.

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

`string.replace(<old>,<new> [,<count>])` заменяет вхождения подстроки в строке

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

`string.startswith([, [,]])` определяет, начинается ли строка с заданной подстроки.

`string.endswith([, [,]])` определяет, заканчивается ли строка заданной подстрокой.

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

`string.isspace()` определяет, состоит ли строка только из пробельных символов.

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

Оператор создает несколько копий строки. Если s это строка, а n целое число, любое из следующих выражений возвращает строку, состоящую из n объединенных копий s :

`s * n`

`n * s`

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

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

31. Как пользоваться методом `partition()` ?

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

`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.

Пример:

```
>>> 'foo.bar'.partition('.')
```

```
('foo', '.', 'bar')
```

```
>>> 'foo@ @bar@ @baz'.partition('@ @')
```

```
('foo', '@ @', 'bar@ @baz')
```

32. В каких ситуациях пользуются методом `rfind()` ?

`.rfind()` , за исключением того, что он вызывает исключение `ValueError` , если `sub` не найден

```
>>> 'Follow Us @Python'.rindex('you')
```

```
Traceback (most recent call last):
```

```
File "", line 1, in 'Follow Us @Python'.rindex('you')
```

```
ValueError: substring not found
```