

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Исследование возможностей Git для работы с
локальными репозиториями»**

**Отчет по лабораторной работе № 2.3
по дисциплине «Основы программной инженерии»**

Выполнил студент группы
ПИЖ-б-о-21-1

Трушева В. О. .« » 2022г.

Подпись студента _____

Работа защищена «
» _____ 20__ г.

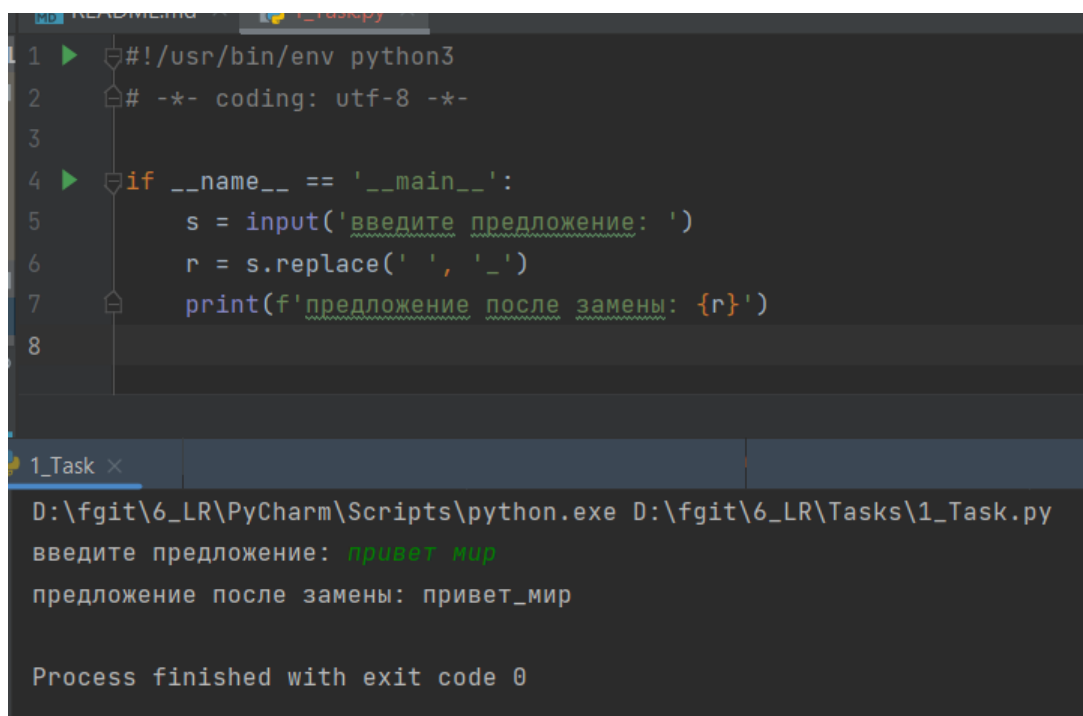
Проверила Воронкин Р.А.

(подпись)

Ставрополь 2022

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.



The screenshot displays the PyCharm IDE interface. The top pane shows a Python script named `1_Task.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
```

The bottom pane shows the execution output for the script:

```
D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Tasks\1_Task.py
введите предложение: привет мир
предложение после замены: привет_мир

Process finished with exit code 0
```

Рисунок 1 – Пример 1

```
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         # длина слова нечетная
10        r = word[:idx]+word[idx+1:]
11    else:
12        # длина слова четная
13        r = word[:idx-1]+word[idx+1:]
14
15    print(r)
16
```

2_Task x

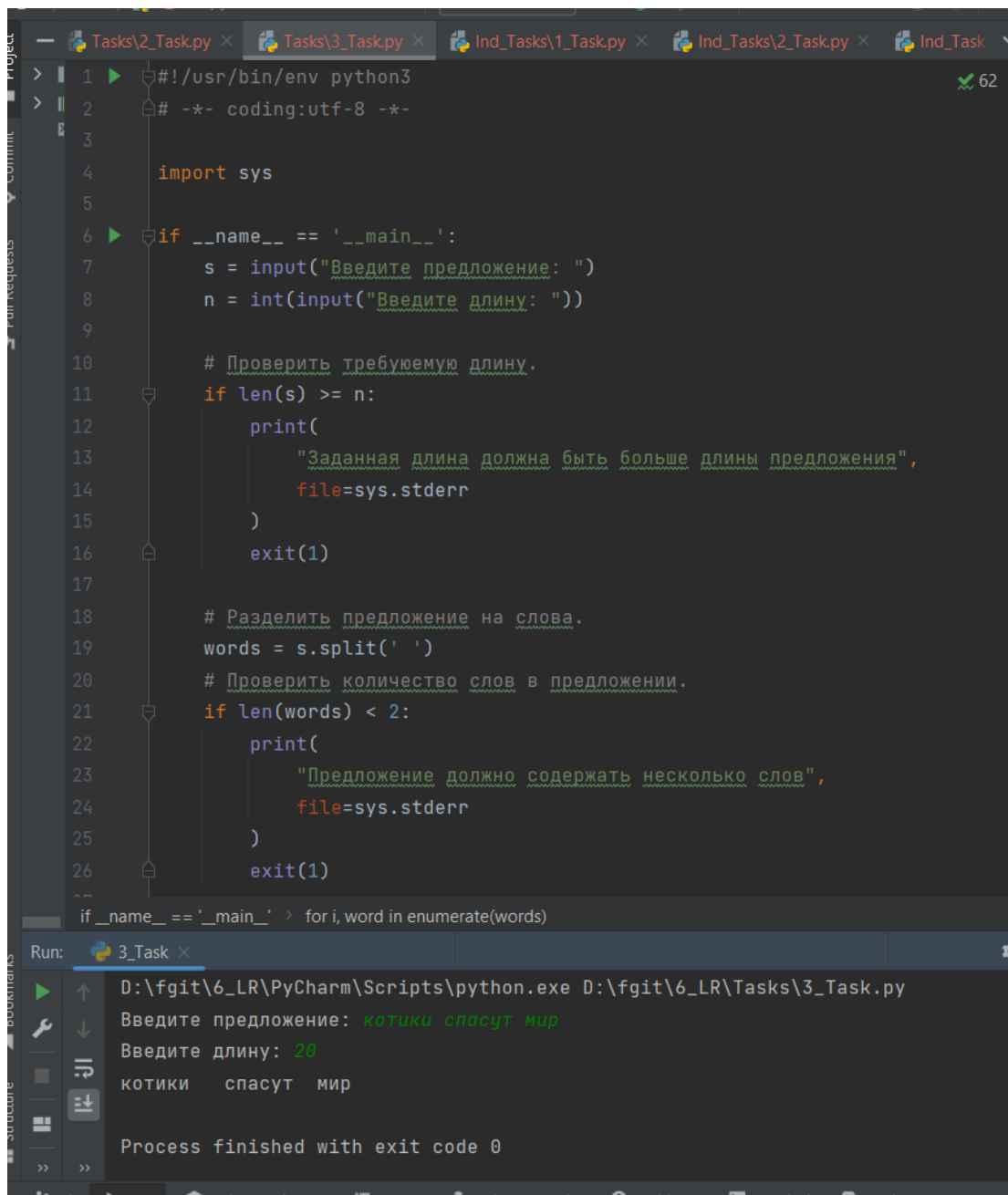
D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Tasks\2_Task.py

введите слово: **привет**

прет

Process finished with exit code 0

Рисунок 2 – Пример 2



The screenshot displays the PyCharm IDE interface. The top pane shows a Python script named `Tasks\3_Task.py` with the following code:

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

import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))

    # Проверить требуемую длину.
    if len(s) >= n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)

    # Разделить предложение на слова.
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
        exit(1)

    if __name__ == '__main__':
        for i, word in enumerate(words)
```

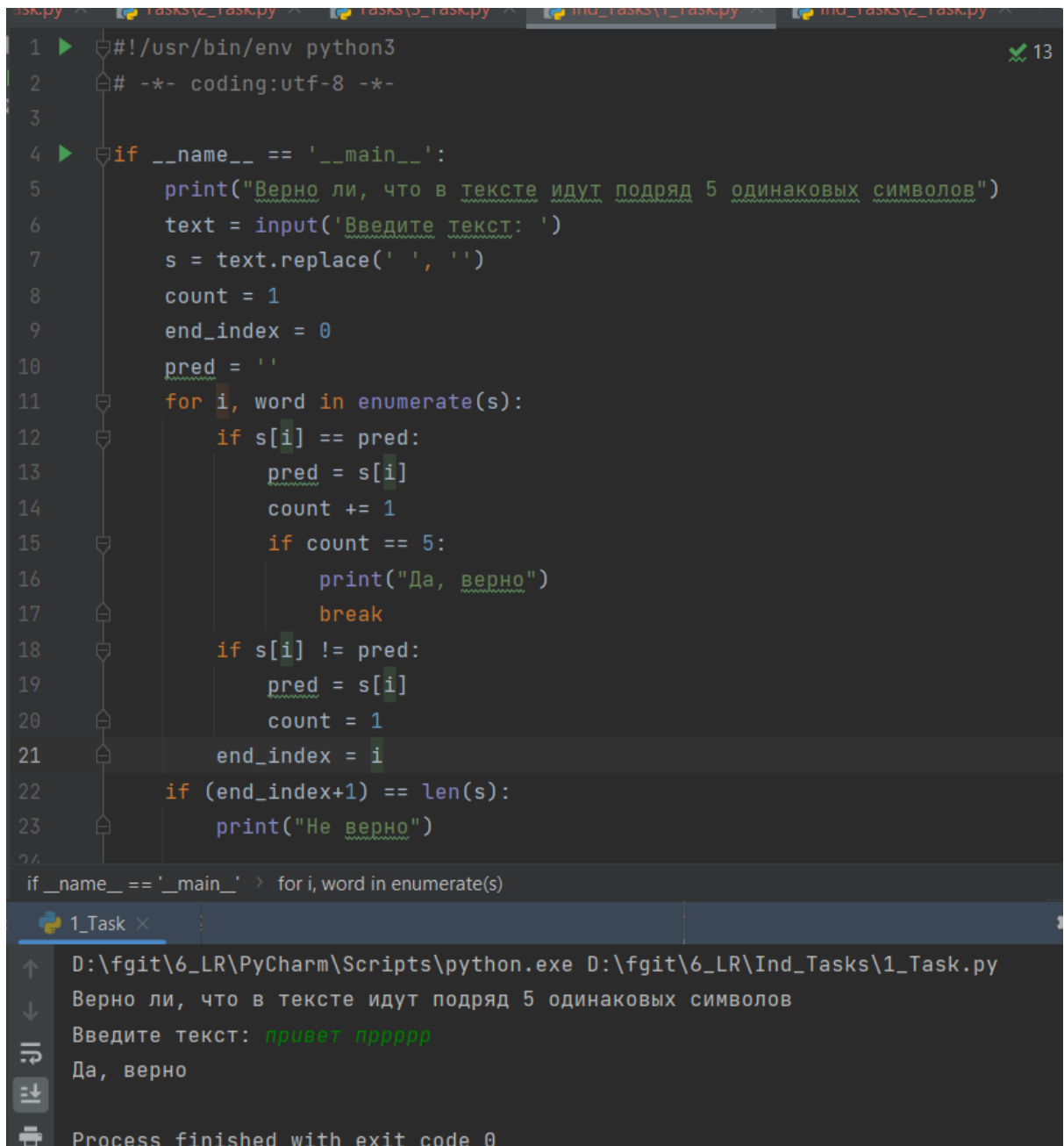
The bottom pane shows the execution output for `3_Task.py`. The command executed is `D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Tasks\3_Task.py`. The input provided is "котика спасут мир" for the sentence and "20" for the length. The output shows the sentence "котика спасут мир" and the message "Process finished with exit code 0".

Рисунок 3 – Пример 3

8. Приведите в отчете скриншоты результатов выполнения каждой из программ примеров при различных исходных данных вводимых с клавиатуры.

9. Выполните индивидуальные задания, согласно своего варианта. Для заданий повышенной сложности номер варианта должен быть получен у преподавателя.

Условие 1 задания. Дан текст. Верно ли, что в нем есть пять идущих подряд одинаковых символов?



```
1 #!/usr/bin/env python3
2 # -*- coding:utf-8 -*-
3
4 if __name__ == '__main__':
5     print("Верно ли, что в тексте идут подряд 5 одинаковых символов")
6     text = input('Введите текст: ')
7     s = text.replace(' ', '')
8     count = 1
9     end_index = 0
10    pred = ''
11    for i, word in enumerate(s):
12        if s[i] == pred:
13            pred = s[i]
14            count += 1
15            if count == 5:
16                print("Да, верно")
17                break
18        if s[i] != pred:
19            pred = s[i]
20            count = 1
21        end_index = i
22    if (end_index+1) == len(s):
23        print("Не верно")
24
if __name__ == '__main__' > for i, word in enumerate(s)
```

1_Task x

D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Ind_Tasks\1_Task.py

Верно ли, что в тексте идут подряд 5 одинаковых символов

Введите текст: *привет ppppp*

Да, верно

Process finished with exit code 0

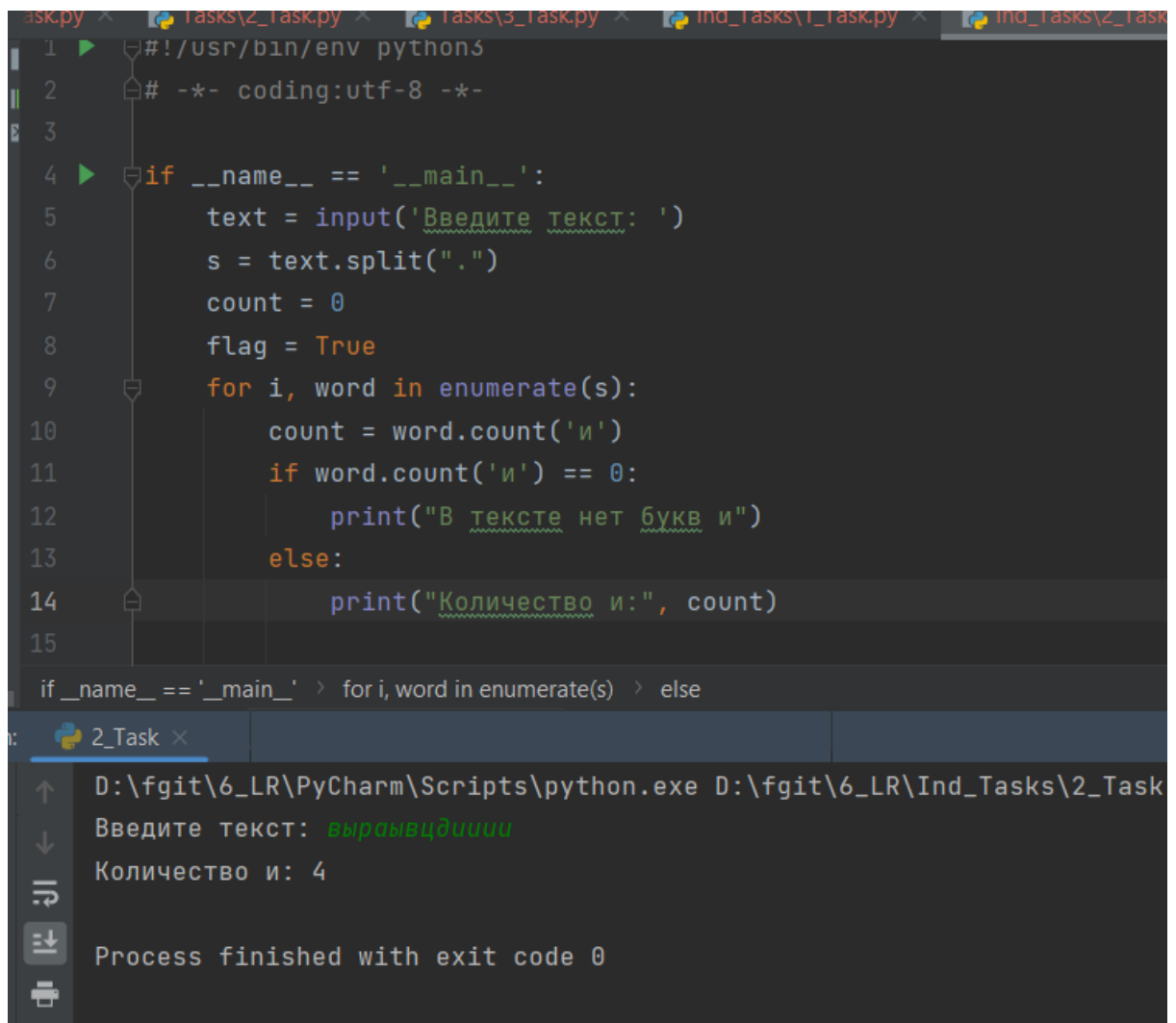
Рисунок 4 – Индивидуальный пример 1

Условие 2 задания. Дан текст. Определить количество букв и в первом предложении. Рассмотреть два случая:

- известно, что буквы и в этом предложении есть;
- букв и в тексте может не быть

```
1  #!/usr/bin/env python3
2  #- coding:utf-8 -*-
3
4  if __name__ == '__main__':
5      text = input('Введите текст: ')
6      s = text.split(".")
7      count = 0
8      flag = True
9      for i, word in enumerate(s):
10         count = word.count('и')
11         if word.count('и') == 0:
12             print("В тексте нет букв и")
13         else:
14             print("Количество и:", count)
15
if __name__ == '__main__' > for i, word in enumerate(s) > else
2_Task x
D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Ind_Tasks\2_Task
Введите текст: fghlsfg
В тексте нет букв и
Process finished with exit code 0
```

Рисунок 5 – Индивидуальный пример 2



```
1 #!/usr/bin/env python3
2 # -*- coding:utf-8 -*-
3
4 if __name__ == '__main__':
5     text = input('Введите текст: ')
6     s = text.split(".")
7     count = 0
8     flag = True
9     for i, word in enumerate(s):
10         count = word.count('и')
11         if word.count('и') == 0:
12             print("В тексте нет букв и")
13         else:
14             print("Количество и:", count)
15
```

2_Task x

D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Ind_Tasks\2_Task

Введите текст: *вырвцдииии*

Количество и: 4

Process finished with exit code 0

Рисунок 6 – Индивидуальный пример 2

Условие 3 задания. Дан текст, в начале которого имеются пробелы и в котором имеются цифры. Найти порядковый номер максимальной цифры, начиная счет с первого символа, не являющегося пробелом. Если максимальных цифр несколько, то должен быть найден номер первой из них.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding:utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     text = input('Введите текст: ')
6     s = text.strip()
7
8     maxi = -1
9     index = -1
10
11     for val in s:
12         if val.isdigit():
13             if int(val) > maxi:
14                 maxi = int(val)
15                 index = s.index(val)
16
17     if maxi == -1 and index == -1:
18         print("В строке нет цифр")
19     else:
20         print(f"Max: {maxi}, index: {index}")
21
```

if __name__ == '__main__' > for val in s > if val.isdigit() > if int(val) > maxi

3_Task x

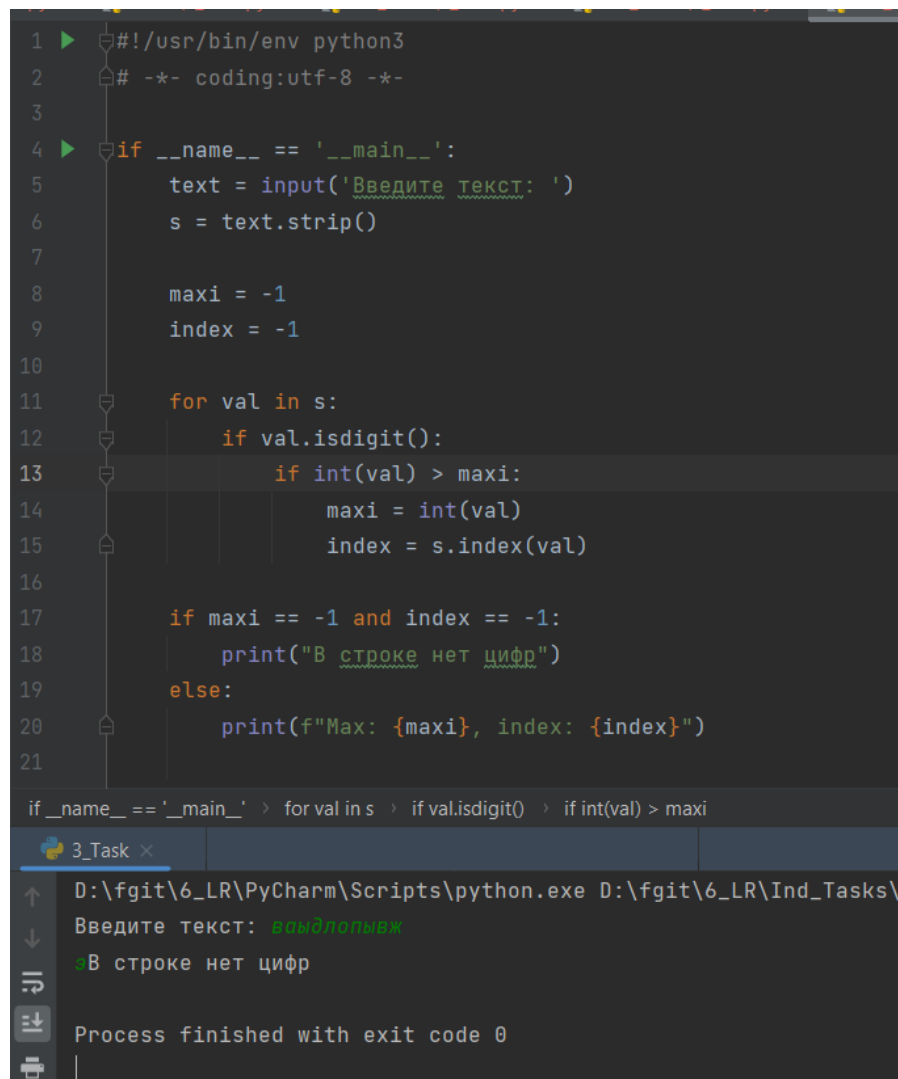
D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Ind_

Введите текст: 7424820

Max: 8, index: 4

Process finished with exit code 0

Рисунок 7 – Индивидуальный пример 3



```
1 #!/usr/bin/env python3
2 # -*- coding:utf-8 -*-
3
4 if __name__ == '__main__':
5     text = input('Введите текст: ')
6     s = text.strip()
7
8     maxi = -1
9     index = -1
10
11     for val in s:
12         if val.isdigit():
13             if int(val) > maxi:
14                 maxi = int(val)
15                 index = s.index(val)
16
17     if maxi == -1 and index == -1:
18         print("В строке нет цифр")
19     else:
20         print(f"Max: {maxi}, index: {index}")
21
```

if __name__ == '__main__' > for val in s > if val.isdigit() > if int(val) > maxi

3_Task x

D:\fgit\6_LR\PyCharm\Scripts\python.exe D:\fgit\6_LR\Ind_Tasks\

Введите текст: **выдлопых**

В строке нет цифр

Process finished with exit code 0

Рисунок 8 – Индивидуальный пример 3

10. Зафиксируйте сделанные изменения в репозитории.
11. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.
12. Выполните слияние ветки для разработки с веткой main / master.
13. Отправьте сделанные изменения на сервер GitHub.
14. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

Вопросы для защиты работы

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, 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 с 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?

```
str.split()
```

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

`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?

`AQ*3 = AQAQAQ`

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>`.