

МИНИСТЕРСТВО 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 кроссплатформенного программирования»

Тема: «Работа со строками в языке Python»

Выполнила: студентка 1 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2022

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

Практическая часть:

1. Создала общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python. Выполнила клонирование созданного репозитория.

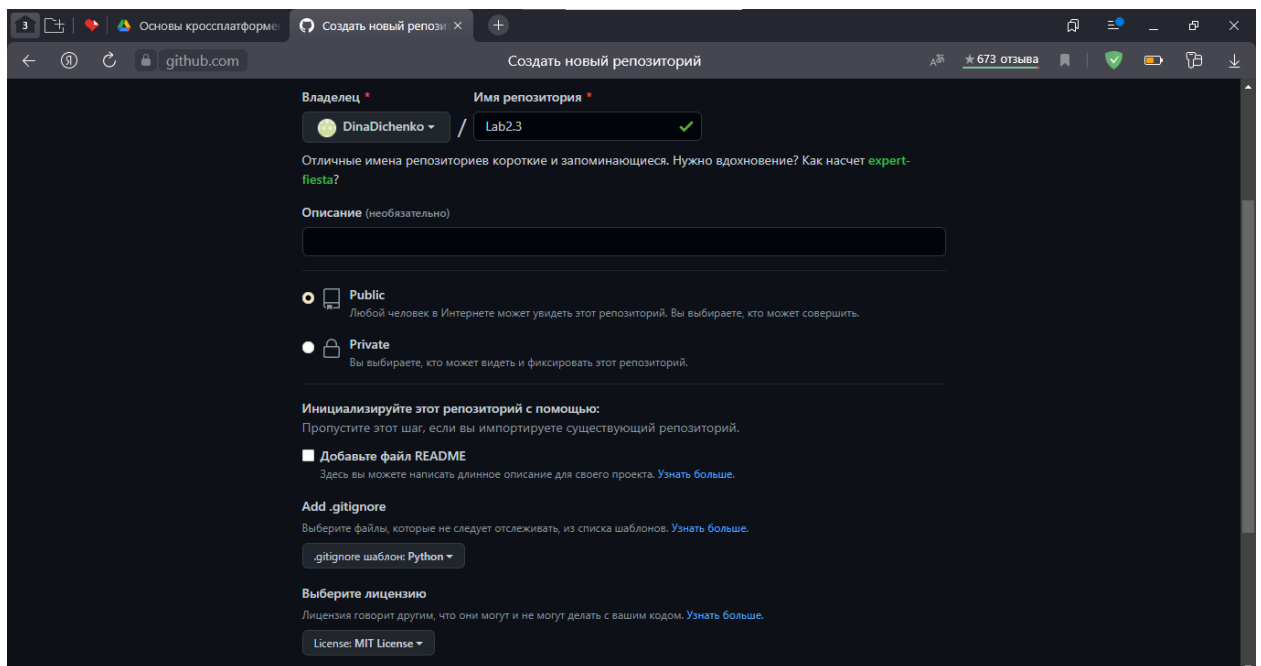


Рисунок 1. Создание репозитория

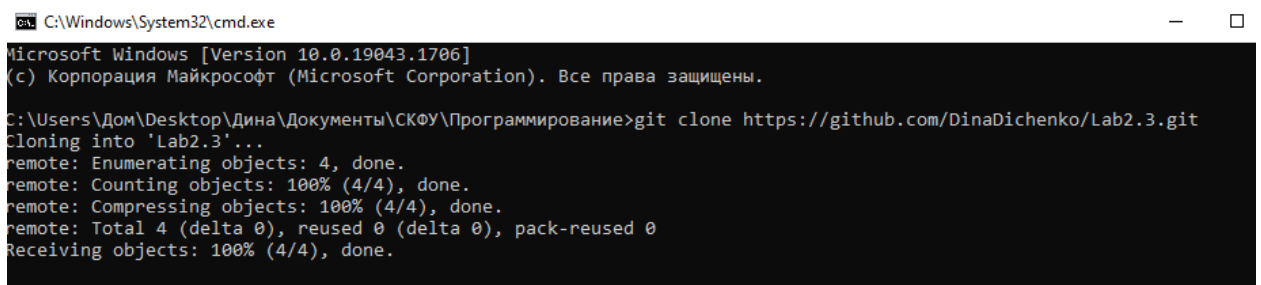


Рисунок 2. Клонирование репозитория

2. Дополнила файл .gitignore необходимыми правилами для работы с IDE PyCharm. Организовала репозиторий в соответствии с моделью ветвления git-flow.

```
Файл  Правка  Формат  Вид  Справка

# Создано https://www.toptal.com/developers/gitignore/api/python , pycharm
# Редактировать по адресу https://www.toptal.com/developers/gitignore?templates=python , pycharm

### PyCharm ###
# Охватывает IDE JetBrains: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm и Rider
# Ссылка: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# Пользовательские вещи
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/словари
.idea/**/полка

# AWS User-specific
.idea/**/aws.xml

# Сгенерированные файлы
.idea/**/contentModel.xml

# Чувствительные или файлы с высоким оттоком
.idea/**/Источники данных/
.idea/**/DataSources.ids
```

Рисунок 3. Изменение файла .gitignore

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.3>git flow init

Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main] develop

Which branch should be used for integration of the "next release"?
- main
Branch name for "next release" development: [] main

How to name your supporting branch prefixes?
Feature branches? [feature/] ft
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/.git/hooks] hook
```

Рисунок 4. Организация репозитория в соответствии с моделью ветвления git-flow.

3. Создала проект PyCharm в папке репозитория. Проработала примеры лабораторной работы. Зафиксировала изменения в репозитории.

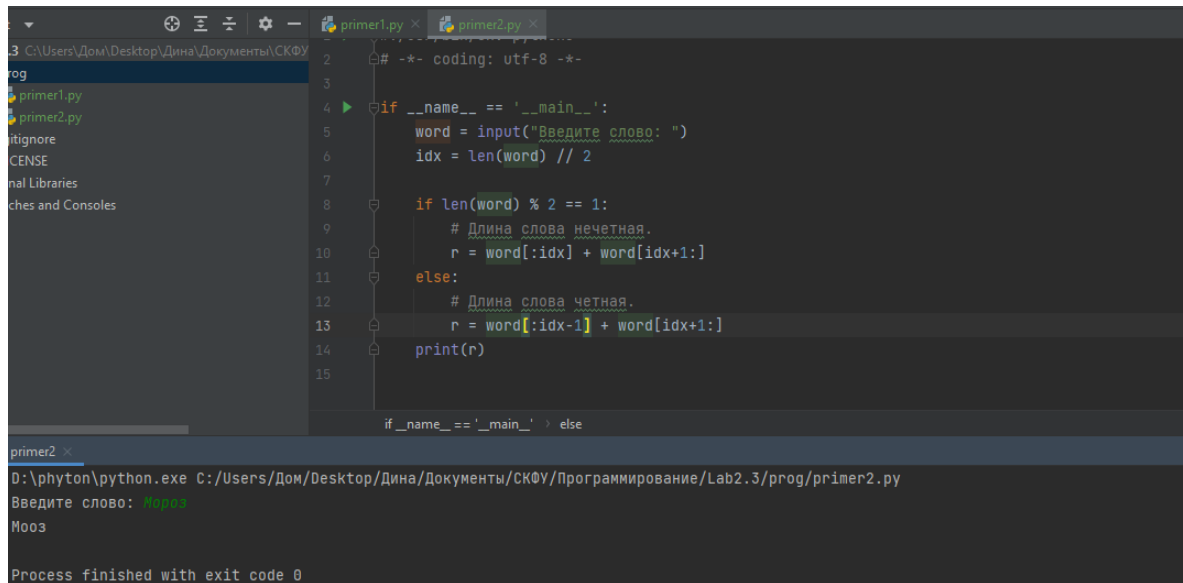
```
23 C:\Users\Дом\Desktop\Дина\Документы\СКФУ> 1 |> #!/usr/bin/env python3
prog 2 |> # -*- coding: utf-8 -*-
primer1.py 3 |>
.gitignore 4 |> if __name__ == '__main__':
LICENSE 5 |> s = input("Введите предложение: ")
6 |> r = s.replace(' ', '_')
7 |> print(f"Предложение после замены: {r}")
8 |>

if __name__ == '__main__':

primer1 x
D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/prog/primer1.py
Введите предложение: My dear friend
Предложение после замены: My_dear_friend

Process finished with exit code 0
```

Рисунок 5. Результат работы программы 1 примера



```
2  -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      word = input("Введите слово: ")
6      idx = len(word) // 2
7
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     print(r)
15
16 if __name__ == '__main__': else
```

primer2 x

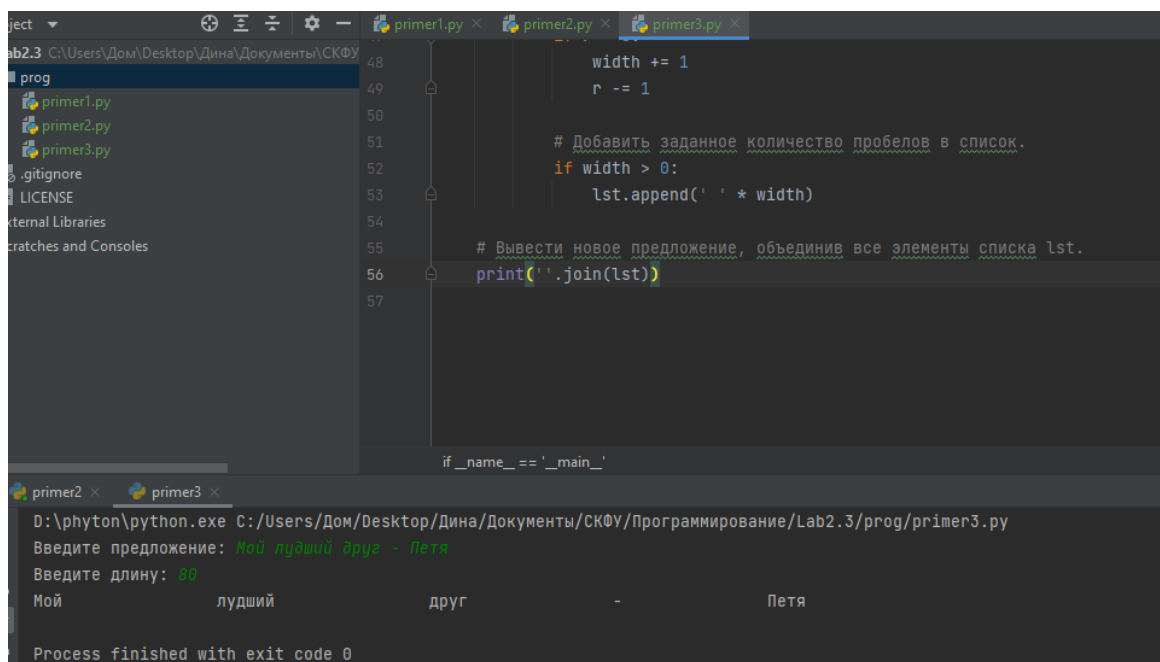
D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/prog/primer2.py

Введите слово: Мооз

Moоз

Process finished with exit code 0

Рисунок 6. Результат работы программы 2 примера



```
48
49     width += 1
50     r -= 1
51
52     # Добавить заданное количество пробелов в список.
53     if width > 0:
54         lst.append(' ' * width)
55
56     # Вывести новое предложение, объединив все элементы списка lst.
57     print(' '.join(lst))
58
59 if __name__ == '__main__':
```

primer2 x primer3 x

D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/prog/primer3.py

Введите предложение: Мой лучший друг - Петя

Введите длину: 80

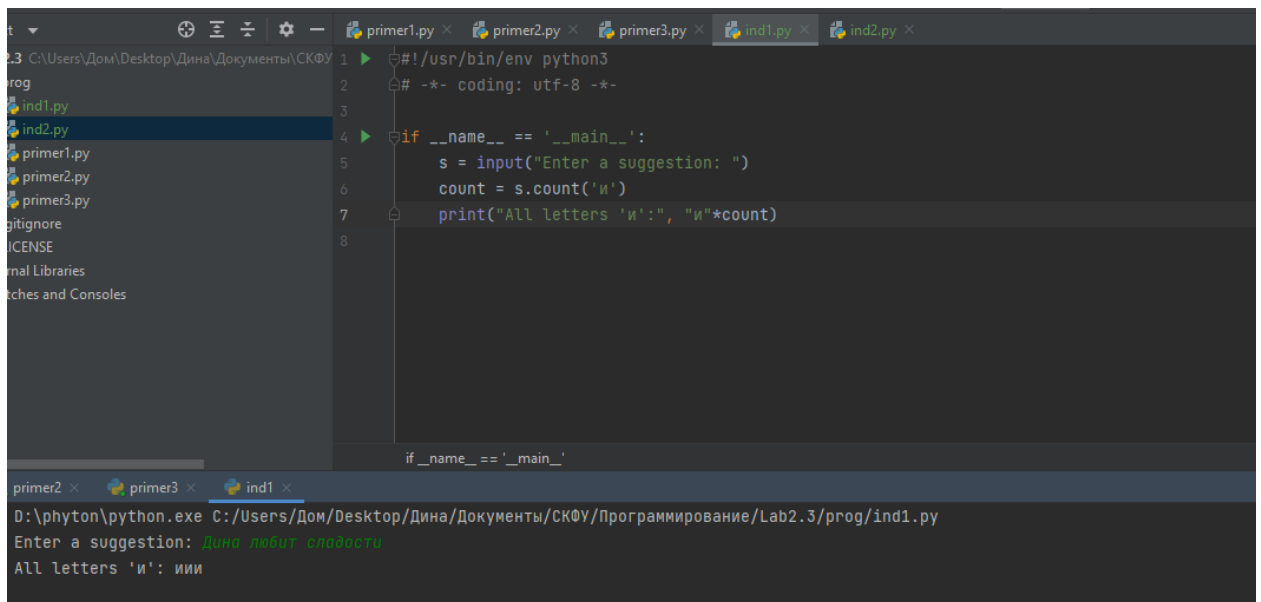
Мой лучший друг - Петя

Process finished with exit code 0

Рисунок 7. Результат работы программы 3 примера

4. Выполнила индивидуальные задания (вариант 7).

Задание 1. Дано предложение. Напечатать все его буквы и.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     s = input("Enter a suggestion: ")
6     count = s.count('и')
7     print("All letters 'и':", "и"*count)
8
```

if __name__ == '__main__'

primer2 x primer3 x ind1 x

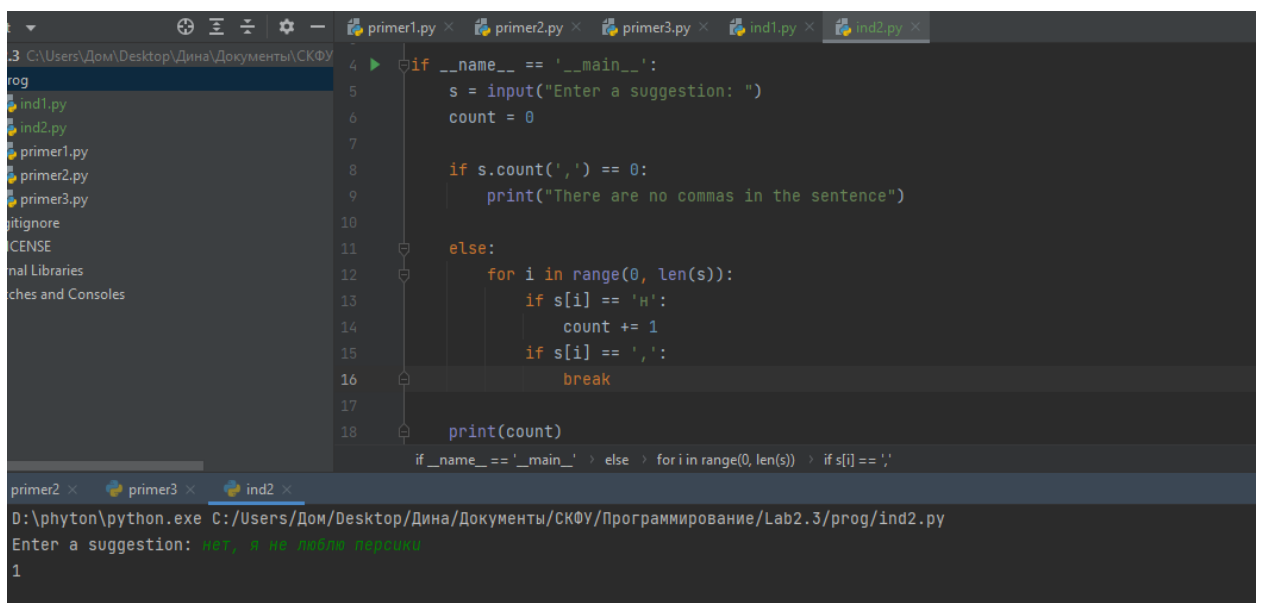
D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/prog/ind1.py

Enter a suggestion: Дина любит сладости

All letters 'и': иии

Рисунок 8. Результат работы программы 1 индивидуального задания

Задание 2. Дано предложение. Определить количество букв н, предшествующих первой запятой предложения. Рассмотреть два случая: известно, что запятые в предложении есть; запятых в предложении может не быть.



```
4 if __name__ == '__main__':
5     s = input("Enter a suggestion: ")
6     count = 0
7
8     if s.count(',') == 0:
9         print("There are no commas in the sentence")
10
11    else:
12        for i in range(0, len(s)):
13            if s[i] == 'н':
14                count += 1
15            if s[i] == ',':
16                break
17
18    print(count)
19
```

if __name__ == '__main__' > else > for i in range(0, len(s)) > if s[i] == ','

primer2 x primer3 x ind2 x

D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/prog/ind2.py

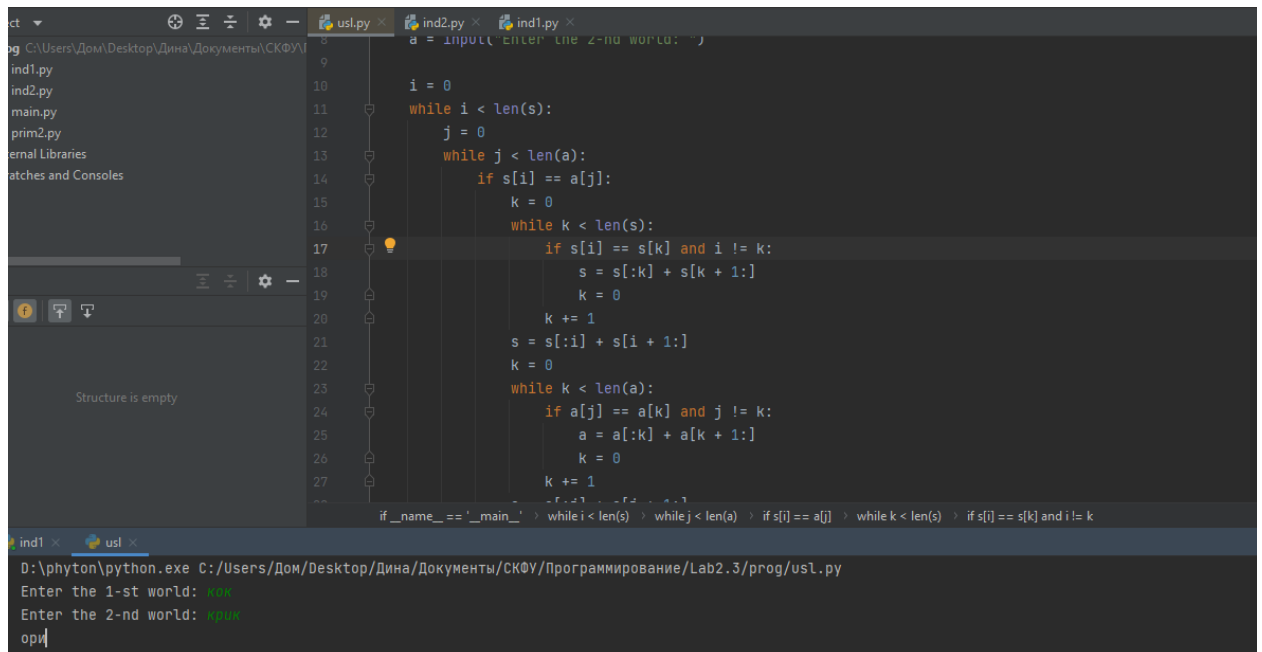
Enter a suggestion: нет, я не люблю персики

1

Рисунок 9. Результат работы программы 2 индивидуального задания

Задание 3. Дано слово. Удалить из него все повторяющиеся буквы, оставив их первые вхождения, т. е. в слове должны остаться только различные буквы.

Задание повышенной сложности. Даны два слова. Напечатать только те буквы слов, которые есть лишь в одном из них (в том числе повторяющиеся). Например, если заданные слова процессор и информация, то ответом должно быть: п е с с и ф м а я.



```
8 a = input('Enter the 1-st world: ')
9
10 i = 0
11 while i < len(s):
12     j = 0
13     while j < len(a):
14         if s[i] == a[j]:
15             k = 0
16             while k < len(s):
17                 if s[i] == s[k] and i != k:
18                     s = s[:k] + s[k + 1:]
19                     k = 0
20                 k += 1
21             s = s[:i] + s[i + 1:]
22             k = 0
23             while k < len(a):
24                 if a[j] == a[k] and j != k:
25                     a = a[:k] + a[k + 1:]
26                     k = 0
27                 k += 1
28             j += 1
29             if j == len(a):
30                 j = 0
31             if j == 0:
32                 i += 1
33         else:
34             j += 1
35     if j == len(a):
36         j = 0
37         i += 1
38     if i == len(s):
39         i = 0
40
41 if __name__ == '__main__':
42     while i < len(s):
43         while j < len(a):
44             if s[i] == a[j]:
45                 while k < len(s):
46                     if s[i] == s[k] and i != k:
47                         s = s[:k] + s[k + 1:]
48                         k = 0
49                     k += 1
50                 s = s[:i] + s[i + 1:]
51                 k = 0
52                 while k < len(a):
53                     if a[j] == a[k] and j != k:
54                         a = a[:k] + a[k + 1:]
55                         k = 0
56                     k += 1
57                 j += 1
58                 if j == len(a):
59                     j = 0
60                 if j == 0:
61                     i += 1
62             else:
63                 j += 1
64         if j == len(a):
65             j = 0
66             i += 1
67         if i == len(s):
68             i = 0
```

ind1 x usl x

D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.3/prog/usl.py

Enter the 1-st world: **rus**

Enter the 2-nd world: **орк**

Рисунок 10. Результат работы программы

Ответы на 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.

3. Какие операции и функции существуют для строк?

Оператор сложения (+), умножения (*), принадлежности подстроки (in).

Функции:

chr() - преобразует целое число в символ;

ord() - преобразует символ в целое число;

len() - возвращает длину строки;

str() - изменяет тип объекта на string.

4. Как осуществляется индексирование строк?

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. 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.

Существует еще один вариант синтаксиса среза, о котором стоит упомянуть. Добавление дополнительного : и третьего индекса означает шаг, который указывает, сколько символов следует пропустить после извлечения каждого символа в срезе.

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.

На самом деле нет особой необходимости изменять строки. Обычно вы можете легко сгенерировать копию исходной строки с необходимыми изменениями. Есть минимум 2 способа сделать это в python.

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

`string.istitle()` определяет, начинаются ли слова строки с заглавной буквы.

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?

Иногда (а точнее, довольно часто) возникают ситуации, когда нужно сделать строку, подставив в неё некоторые данные, полученные в процессе

выполнения программы (пользовательский ввод, данные из файлов и т. д.). Подстановку данных можно сделать с помощью форматирования строк. Форматирование можно сделать с помощью оператора % , либо с помощью метода format , либо с помощью так называемых f-строк. Форматирование с помощью оператора % относится к устаревшим способам форматирования, поэтому в рамках данной лабораторной работы будут рассмотрены только метод format и f-строки.

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?

`string.isalpha().`

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

`string.istitle()` определяет, начинаются ли слова строки с заглавной буквы.

`s.istitle()` возвращает True когда s не пустая строка и первый алфавитный символ каждого слова в верхнем регистре, а все остальные буквенные символы в каждом слове строчные. Возвращает False , если нет.

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 и последний символы 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()`?

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

Она напечатается 3 раза.

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

`s.title()`.

31. Как пользоваться методом `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>`.

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