

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №4

Работа со строками в языке Python

Выполнил студент группы ИТС-б-з-22-1

Дубинин Олег Александрович

« » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 2023г.

Проверил доцент, кандидат технических
наук, доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

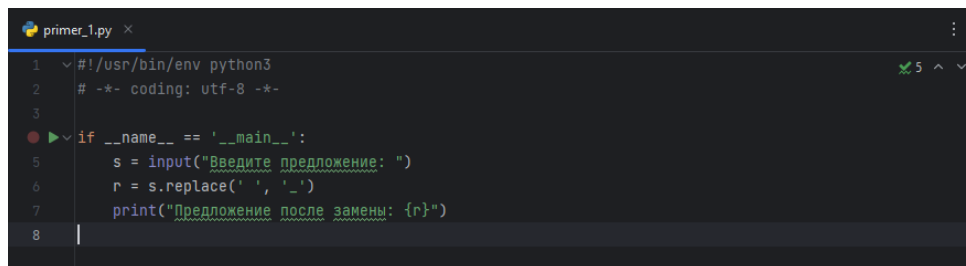
Ставрополь, 2023 г.

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

Ход работы:

<https://github.com/OLEG1232155/4> - репозиторий

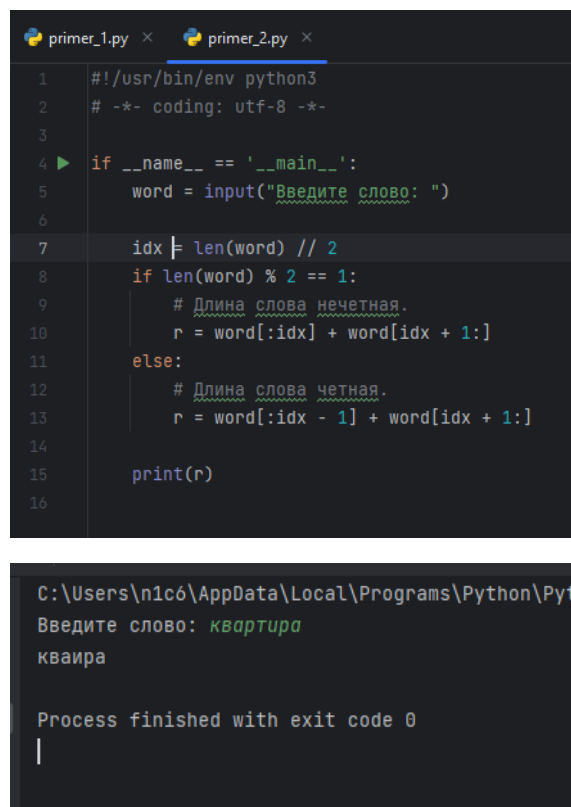
Пример 1. Дано предложение. Все пробелы в нем заменить символом «_».



```
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("Предложение после замены: {r}")
8
```

Рисунок 1 – Окно программы первого примера

Пример 2. Дано слово. Если его длина нечетная, то удалить среднюю букву, в противном случае – две средние буквы.



```
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
```

C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe

Введите слово: квартира

кваира

Process finished with exit code 0

Рисунок 2 – Окно программы второго примера

Пример 3. Дана строка текста, в котором нет начальных и конечных пробелов. Необходимо изменить ее так, чтобы длина строки стала равна заданной длине (предполагается, что требуемая длина не меньше исходной). Это следует сделать путем вставки между словами дополнительных пробелов. Количество пробелов между отдельными словами должно отличаться не более чем на 1.

```
primer_1.py  primer_2.py  primer_3.py ×
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      s = input("Введите предложение: ")
8      n = int(input("Введите длину: "))
9
10     # Проверить требуемую длину.
11     if len(s) >= n:
12         print(
13             "Заданная длина должна быть больше длины предложения",
14             file=sys.stderr
15         )
16         exit(1)
17
18     # Разделить предложение на слова.
19     words = s.split(' ')
20     # Проверить количество слов в предложении.
21     if len(words) < 2:
22         print(
23             "Предложение должно содержать несколько слов",
24             file=sys.stderr
25         )
26         exit(1)
27
28     # Количество пробелов для добавления.
29     delta = n
30     for word in words:
31         delta -= len(word)
32
33     # Количество пробелов на каждое слово.
34     w, r = delta // (len(words) - 1), delta % (len(words) - 1)
35
36     # Сформировать список для хранения слов и пробелов.
37     lst = []
38
39     # Пронумеровать все слова в списке и перебрать их.
40     for i, word in enumerate(words):
41         lst.append(word)
42
43         # Если слово не является последним, добавить пробелы.
44         if i < len(words) - 1:
45             # Определить количество пробелов.
46             width = w
47             if r > 0:
48                 width += 1
49                 r -= 1
50
51             # Добавить заданное количество пробелов в список.
52             if width > 0:
53                 lst.append(' ' * width)
54
55     # Вывести новое предложение, объединив все элементы списка lst.
56     print(''.join(lst))
```

```
C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\n1c6\
Введите предложение: живопись очень красивая
Введите длину: 40
живопись          очень          красивая

Process finished with exit code 0
|
```

Рисунок 3 – Окно программы третьего примера

Первое индивидуальное задание.

8. Дано предложение. Составить программу, которая печатает «столбиком» все вхождения в предложение некоторого символа.

```
individ_1.py x individ_2.py individ_3.py pov.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = input("Введите предложение: ")
6      a = input("Введите символ: ")
7
8      k = s.count(a)
9
10
11     if k == 0:
12         print("В предложении нет данного символа.")
13     else:
14         i = 1
15         while i <= k:
16             print(a)
17             i += 1
```

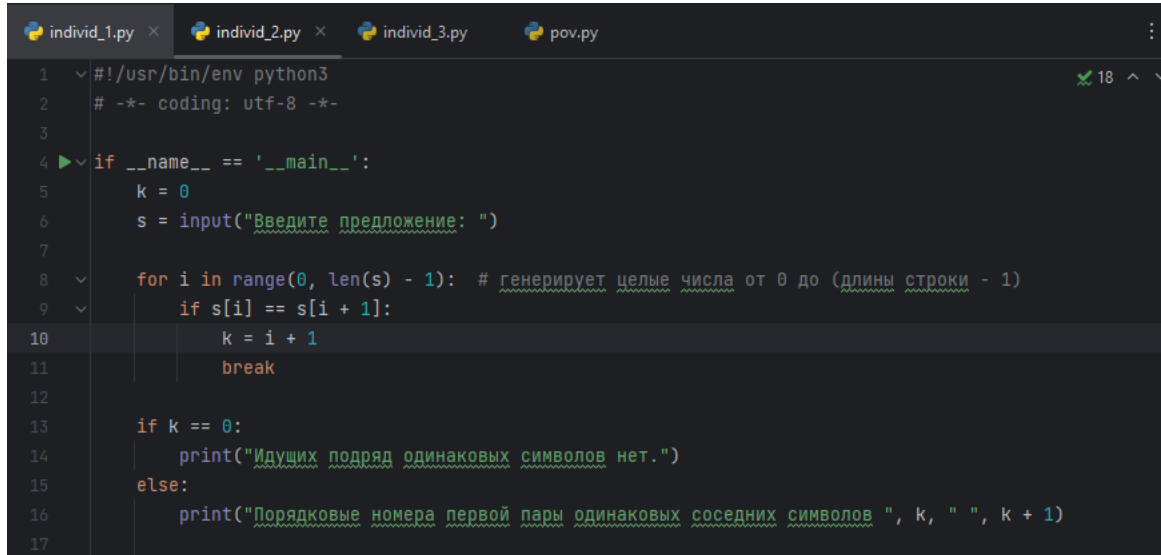
```
C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe "C:
Введите предложение: я живу хорошо
Введите символ: o
o
o
o

Process finished with exit code 0
```

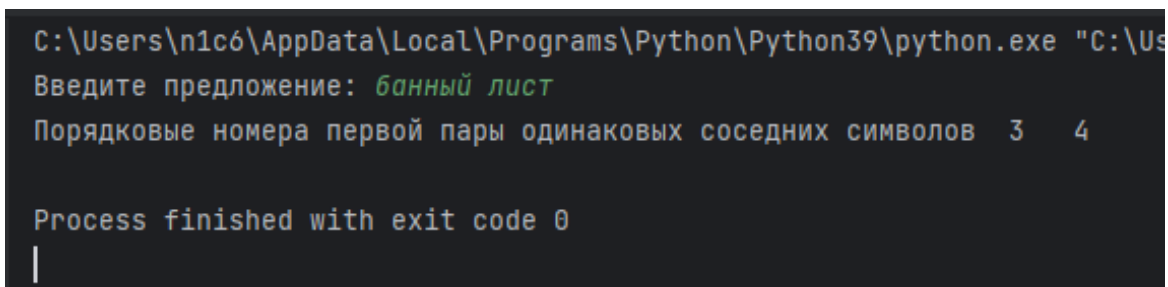
Рисунок 4 – Окно программы для первой задачи и проверка работоспособности кода.

Второе индивидуальное задание.

8. Дано предложение. Определить порядковые номера первой пары одинаковых соседних символов. Если таких символов нет, то должно быть напечатано соответствующее сообщение.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      k = 0
6      s = input("Введите предложение: ")
7
8      for i in range(0, len(s) - 1): # генерирует целые числа от 0 до (длины строки - 1)
9          if s[i] == s[i + 1]:
10             k = i + 1
11             break
12
13     if k == 0:
14         print("Идущих подряд одинаковых символов нет.")
15     else:
16         print("Порядковые номера первой пары одинаковых соседних символов ", k, " ", k + 1)
```



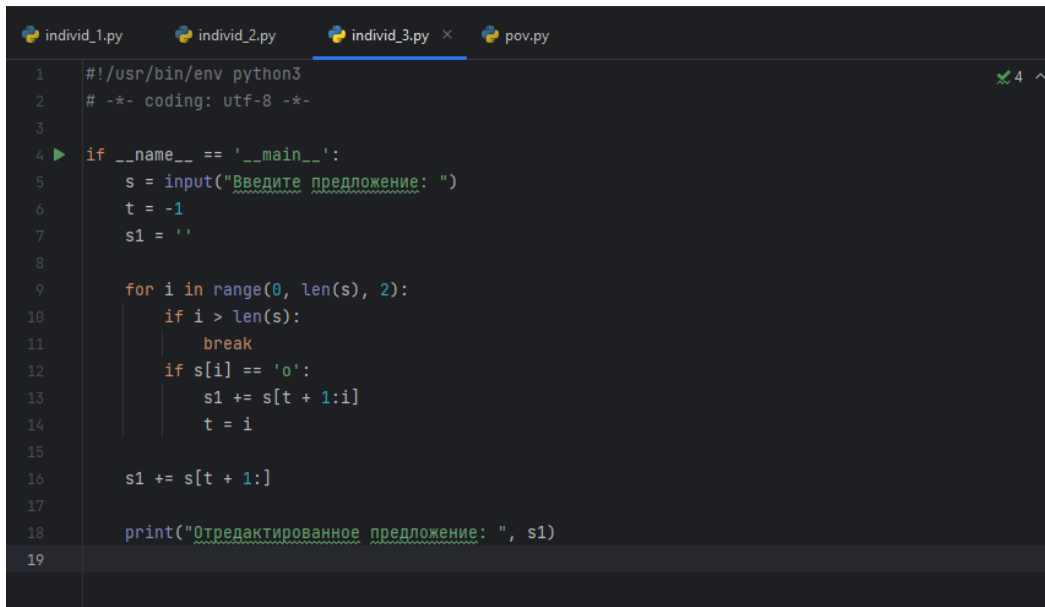
```
C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe "C:\Us
Введите предложение: банный лист
Порядковые номера первой пары одинаковых соседних символов 3 4

Process finished with exit code 0
|
```

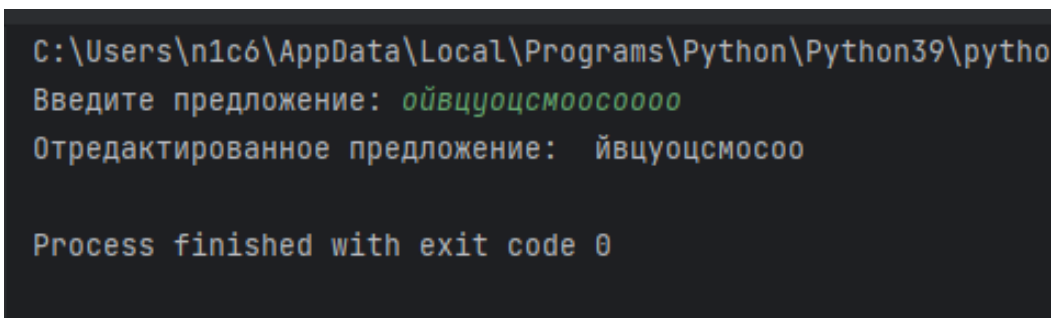
Рисунок 5 – Окно программы для второй задачи и проверка работоспособности кода.

Третье индивидуальное задание.

8. Дано предложение. Удалить из него все буквы о, стоящие на нечетных местах.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = input("Введите предложение: ")
6      t = -1
7      s1 = ''
8
9      for i in range(0, len(s), 2):
10         if i > len(s):
11             break
12         if s[i] == 'o':
13             s1 += s[t + 1:i]
14             t = i
15
16     s1 += s[t + 1:]
17
18     print("Отредактированное предложение: ", s1)
```



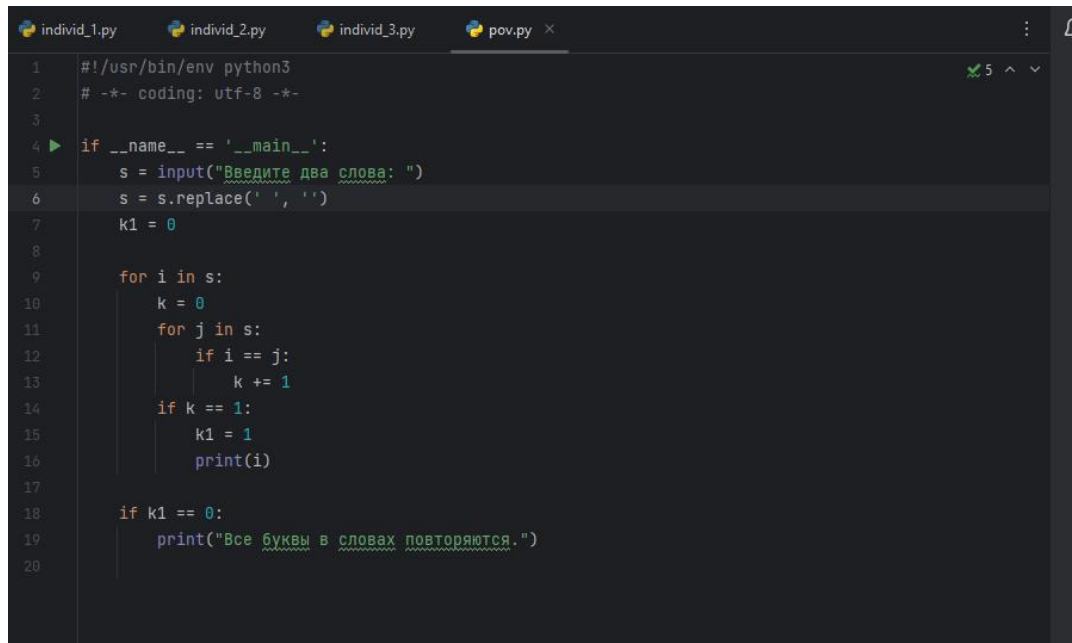
```
C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe
Введите предложение: ойвцуоцсмоосооооо
Отредактированное предложение: йвцуоцсмооо

Process finished with exit code 0
```

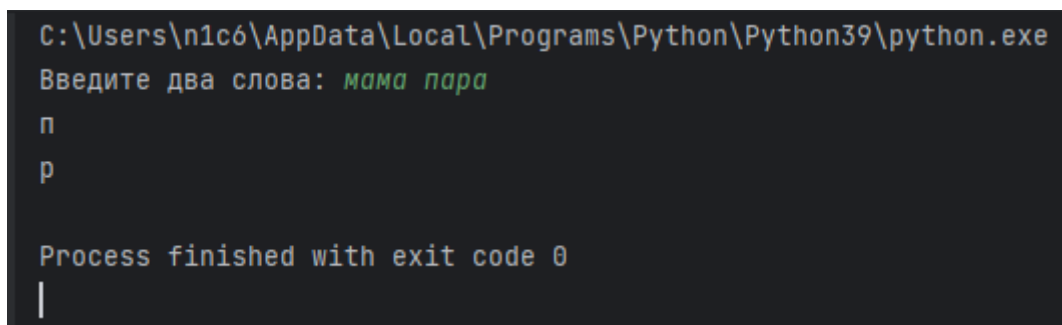
Рисунок 6 – Окно программы для третьей задачи и проверка работоспособности кода.

Повышенная сложность.

8. Даны два слова. Напечатать только те буквы слов, которые встречаются в обоих словах лишь один раз. Например, если заданные слова процессор и информация, то ответом должно быть: п е ф м а я.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = input("Введите два слова: ")
6      s = s.replace(' ', '')
7      k1 = 0
8
9      for i in s:
10         k = 0
11         for j in s:
12             if i == j:
13                 k += 1
14         if k == 1:
15             k1 = 1
16             print(i)
17
18     if k1 == 0:
19         print("Все буквы в словах повторяются.")
20
```



```
C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe
Введите два слова: мама пара
п
р

Process finished with exit code 0
|
```

Рисунок 7 – Окно программы для задания повышенной сложности и проверка работоспособности кода.

Вывод: приобрел навыки по работе со строками при написании программ с помощью языка программирования Python.