**Федеральное государственное образовательное**

**бюджетное учреждение высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**ИНСТИТУТ ЗАОЧНОГО И ОТКРЫТОГО ОБРАЗОВАНИЯ**

**ДЕПАРТАМЕНТ АНАЛИЗА ДАННЫХ,**

**ПРИНЯТИЯ РЕШЕНИЙ И ФИНАНСОВЫХ ТЕХНОЛОГИЙ**

**ПРАКТИЧЕСКАЯ РАБОТА № 2**

**по дисциплине «Алгоритмы и структуры данных в языке Python»**

**Тема: строки**

Направление подготовки - 09.03.03 «Прикладная информатика»

Студент(ка) **Белоусов А.А.**

(Ф.И.О.)

Курс1 № группы **ЗБ-ПИ1-1**

Преподаватель **Петросов Д.А.**

(Ф.И.О.)

Москва 2020

Изучение теории в Jupiter Notebook

На рисунках 1-15 представлены результаты выполнения обучающих примеров.

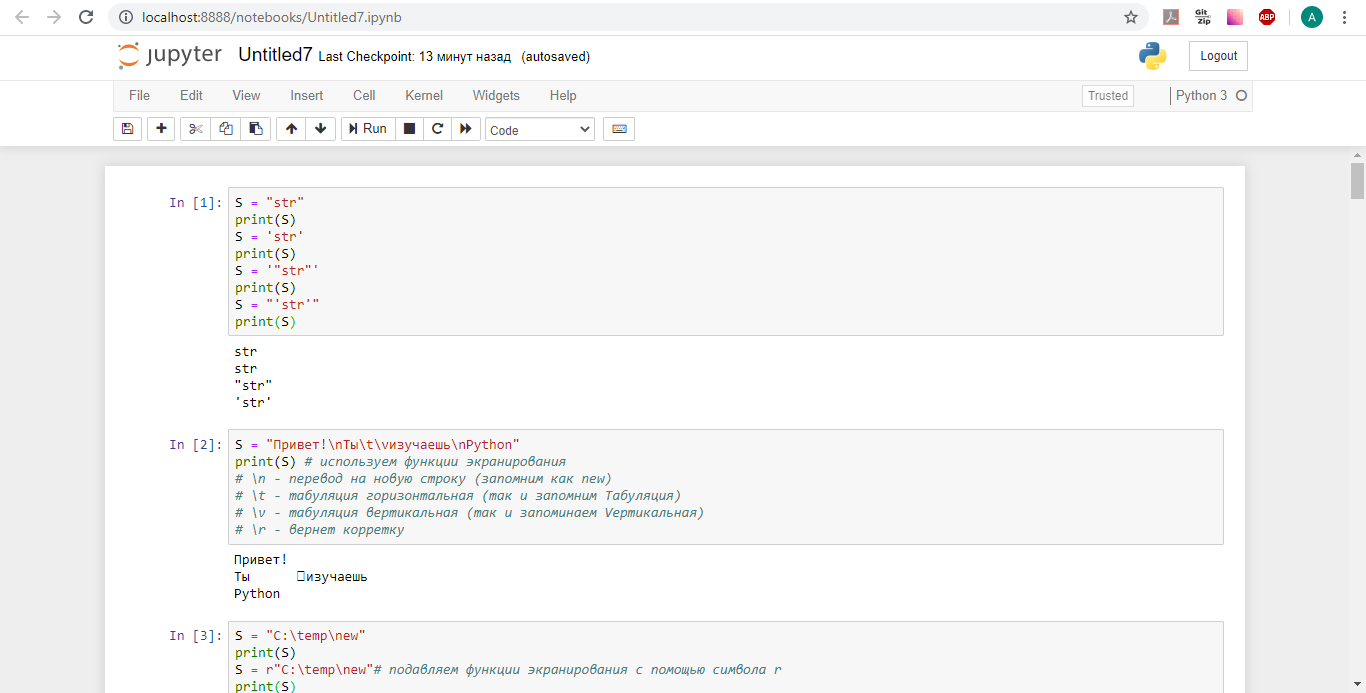


Рис.1.

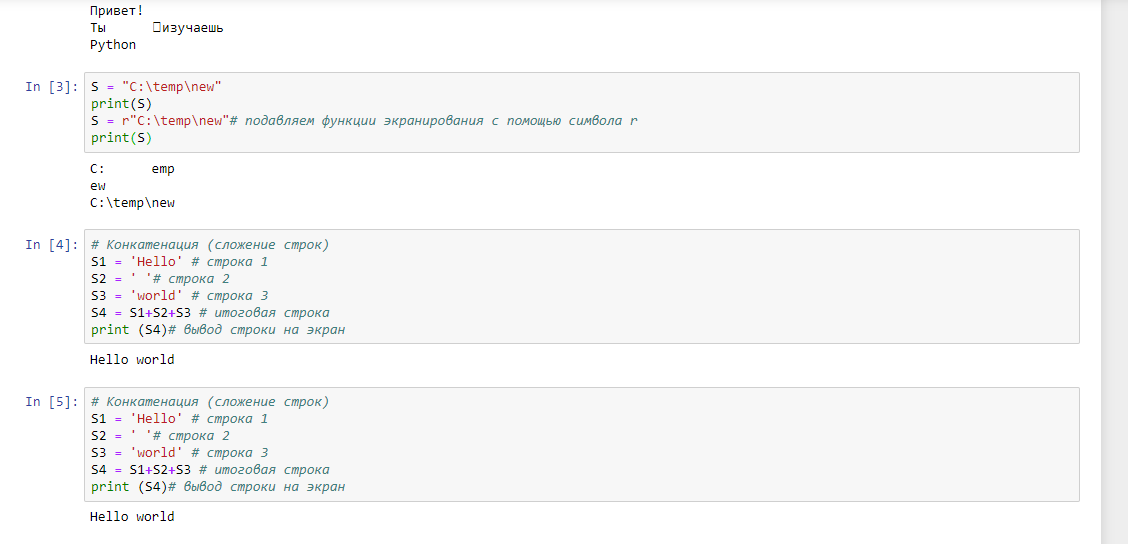


Рис.2.

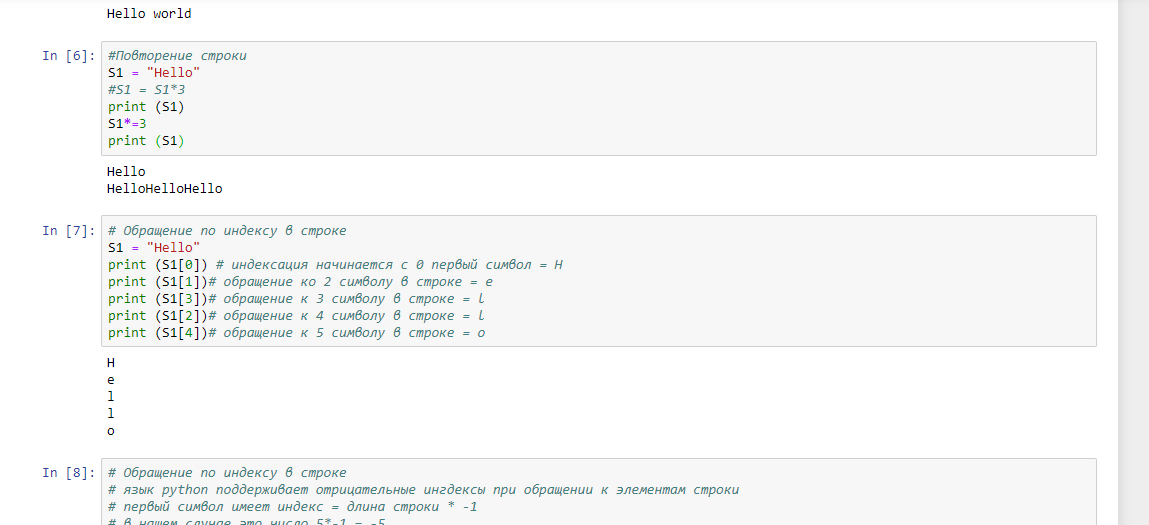


Рис.3.

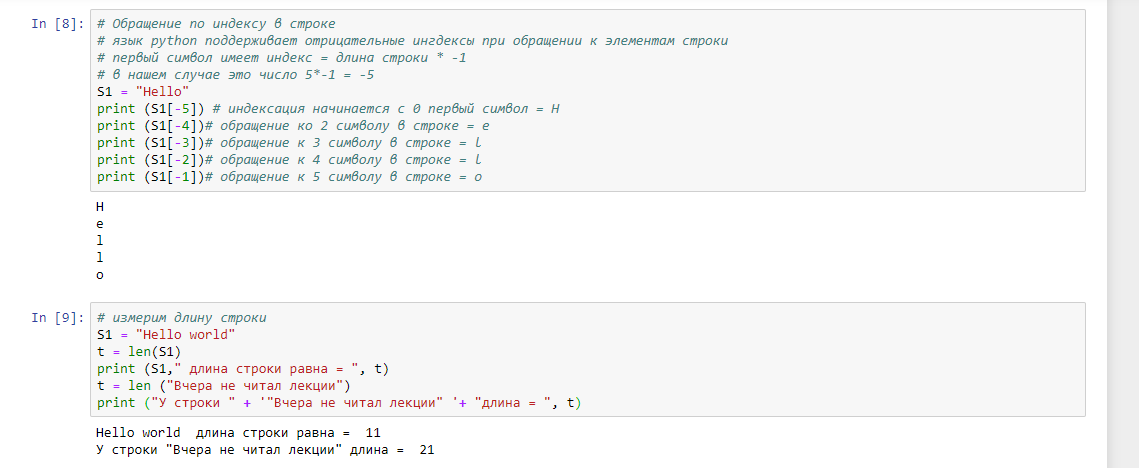


Рис.4.

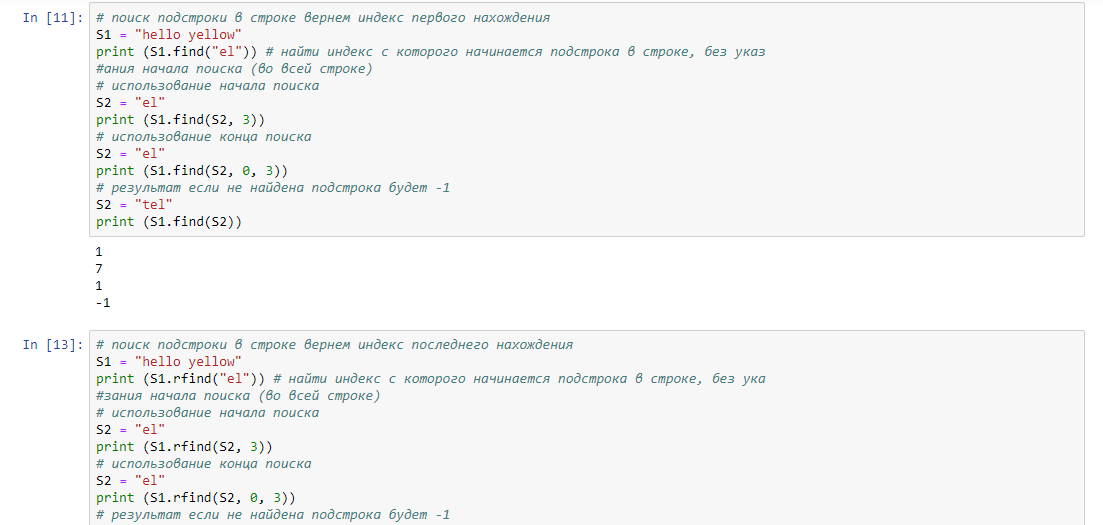


Рис.5.

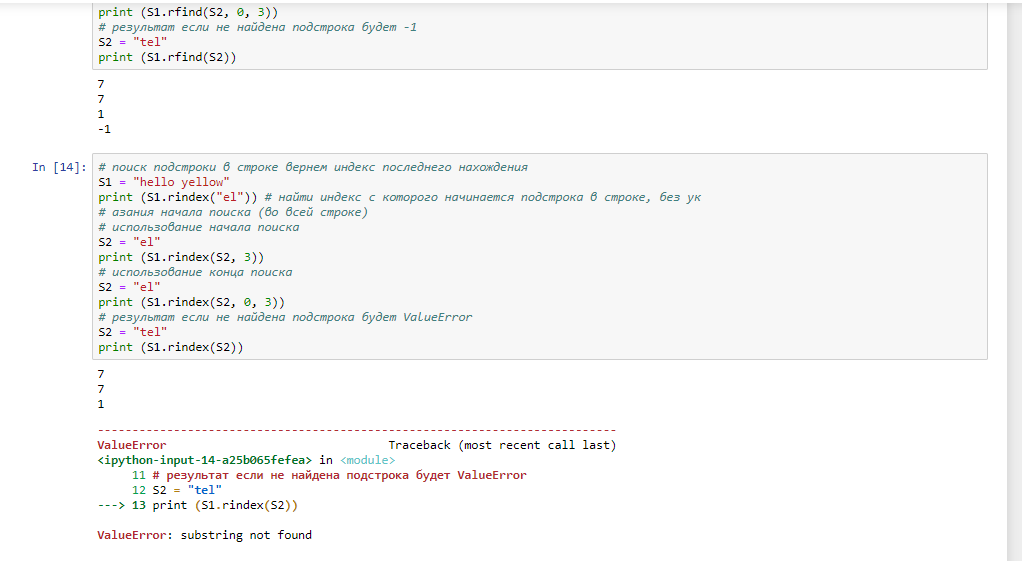


Рис.6.

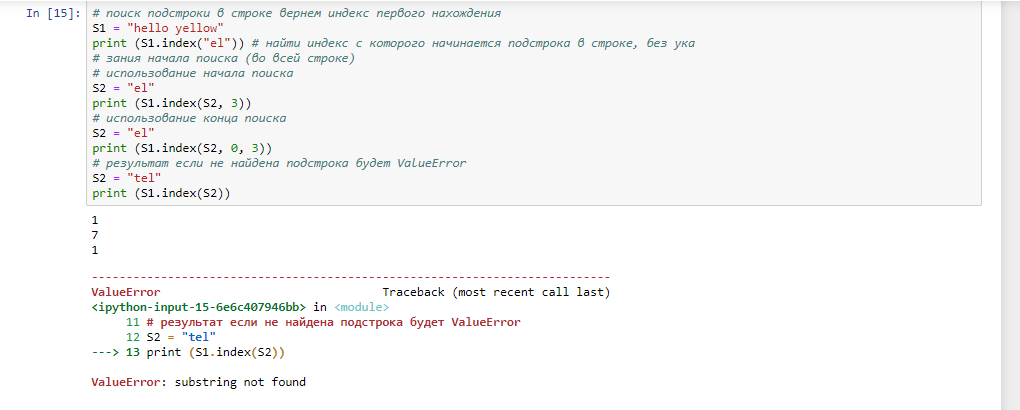


Рис.7.



Рис.8.

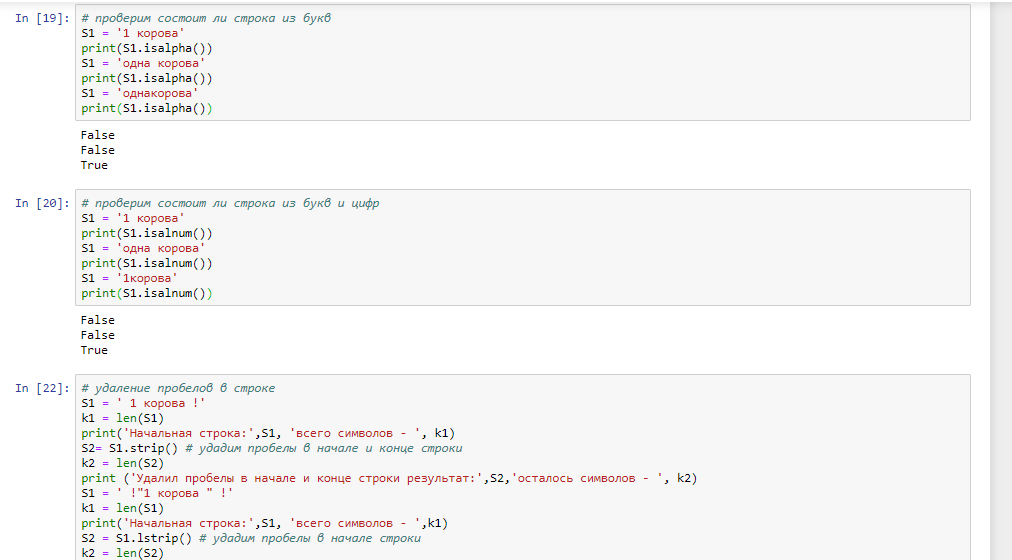


Рис.9.

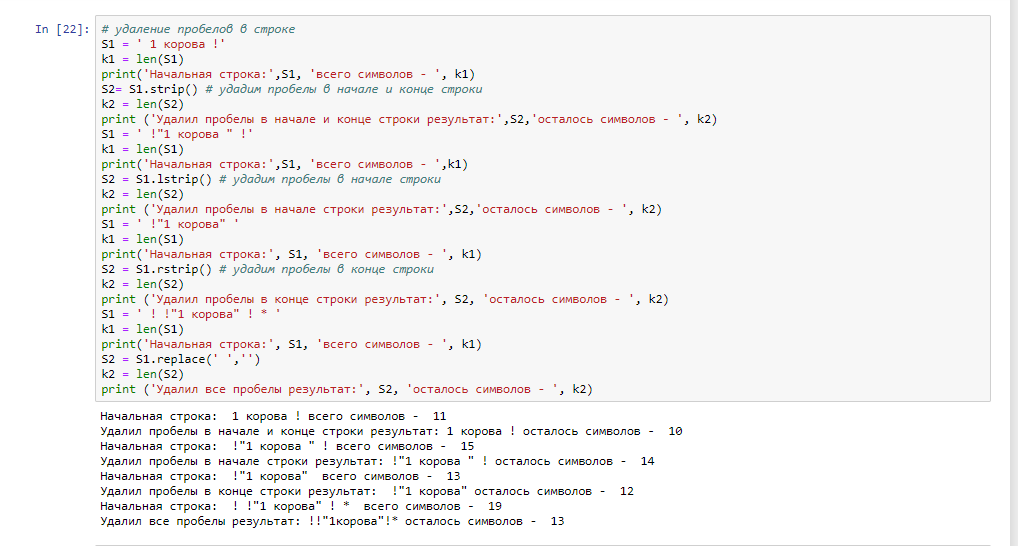


Рис.10.

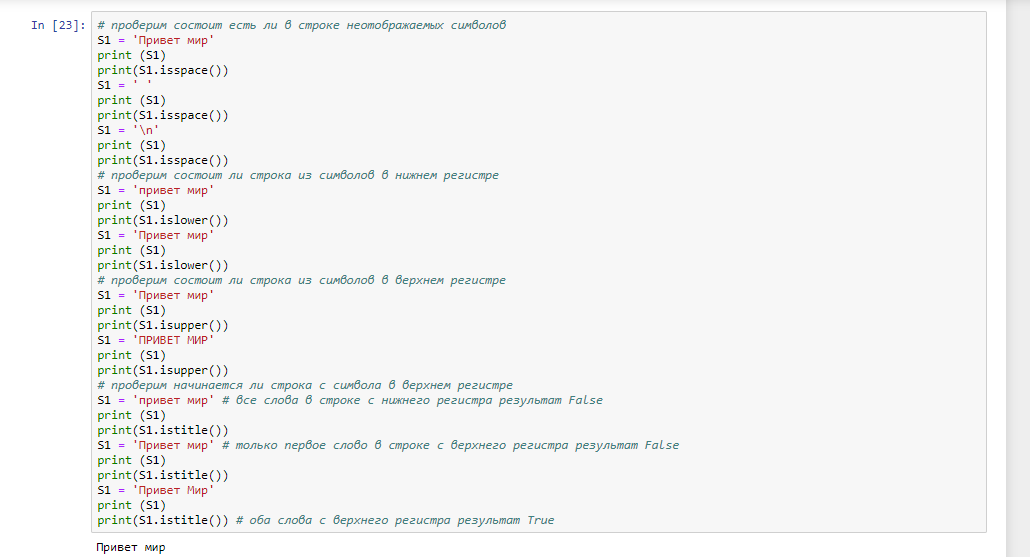


Рис.11.

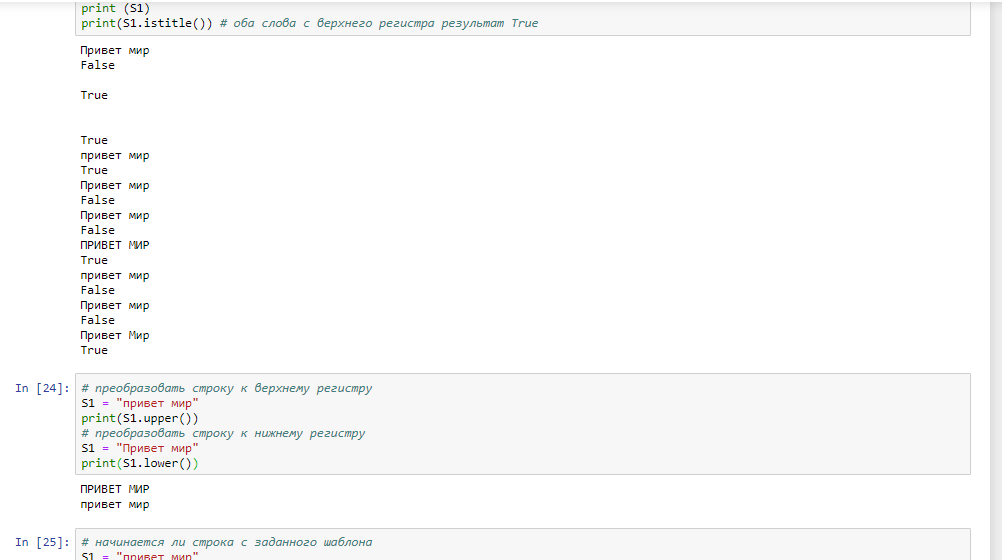


Рис.12.



Рис.13.

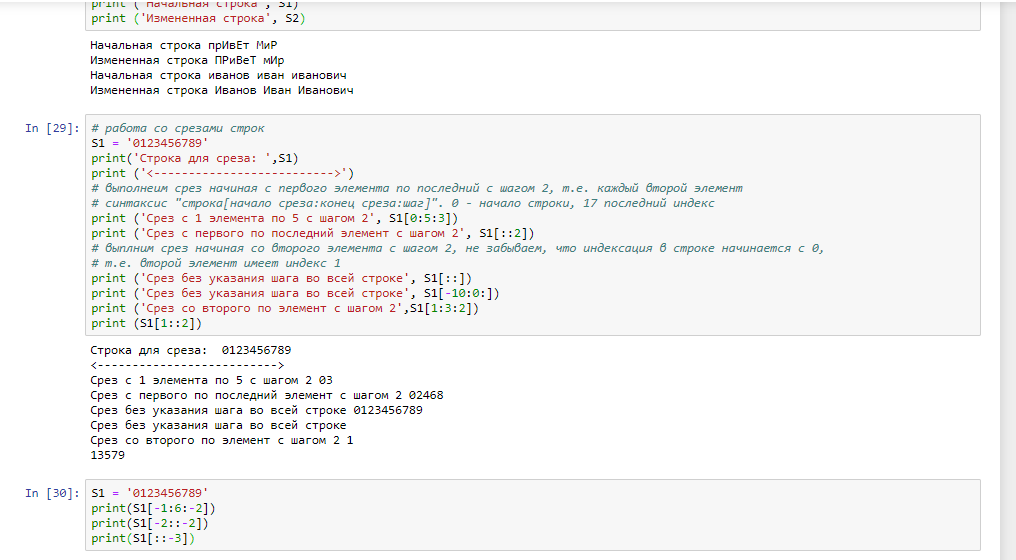


Рис.14.

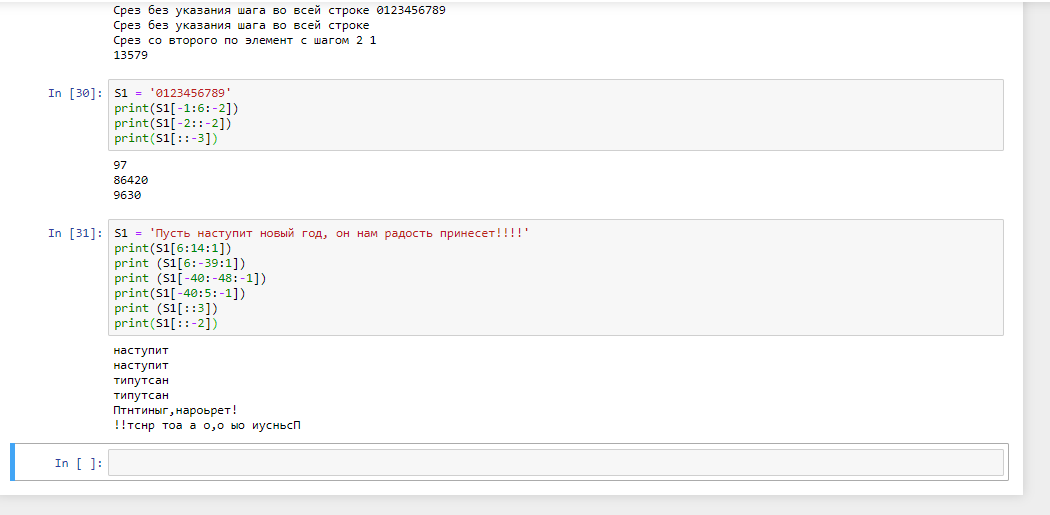


Рис.15.

Задания

**Задание 1**

Реализовать одно алфавитный шифр Цезаря для шифрования и дешифрование строки любой длины и заданным ключем (сдвигом алфавита), используется кириллический алфавит, знаки препинания, цифры, верхний и нижний регистр.

**Задание 2**

Реализовать шифр с использованием кодового слова, используется латинский алфавит с верхним регистром.

**Задание 3**

Реализовать двух алфавитный шифр Цезаря для шифрования и дешифрование строки любой длины и заданным ключем, используется латинский алфавит и цифры, а так же только нижний регистр.

**Задание 4**

Реализовать шифр Виженера, который состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Строка для шифрования должна быть на основе латинского алфавита, ключевое слово вводится с клавиатуры.

Код программы

**from** collections **import** OrderedDict  
**import** string  
  
  
*# Задание 1***def** toCesar(str, k):  
 alphabetUp = **"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"** alphabetLower = **"абвгдеёжзийклмнопрстуфхцчшщъыьэюя"** result = **""** k %= len(alphabetUp)  
  
 **for** char **in** str:  
 i = alphabetUp.find(char)  
 **if** i != -1:  
 result += alphabetUp[(i + k) % len(alphabetUp)];  
 **continue** i = alphabetLower.find(char)  
 **if** i != -1:  
 result += alphabetLower[(i + k) % len(alphabetLower)];  
 **continue** result += char  
  
 **return** result  
  
  
**def** fromCesar(str, k):  
 lenalpha = 33  
 **return** toCesar(str, lenalpha - k % lenalpha)  
  
  
*# Задание 2***def** toCode(str, codeword):  
 alphabet = string.ascii\_uppercase  
 secretalpha = list(OrderedDict.fromkeys(codeword.upper() + alphabet).keys())  
 result = **""  
  
 for** char **in** str:  
 i = alphabet.find(char)  
 **if** i != -1:  
 result += secretalpha[i]  
 **continue** result += char  
  
 **return** result  
  
  
**def** fromCode(str, codeword):  
 alphabet = **''**.join(list(OrderedDict.fromkeys(codeword.upper() + string.ascii\_uppercase).keys()))  
 secretalpha = string.ascii\_uppercase  
 result = **""  
  
 for** char **in** str:  
 i = alphabet.find(char)  
 **if** i != -1:  
 result += secretalpha[i]  
 **continue** result += char  
  
 **return** result  
  
*# задание 3***def** toDoubleCesar(str, k1, k2):  
 alphabet = string.ascii\_lowercase+string.digits  
 result = **""** k1 %= len(alphabet)  
 k2 %= len(alphabet)  
  
 **for** i **in** range(len(str)):  
 char = str[i]  
 index = alphabet.find(char)  
 **if** index != -1:  
 result += alphabet[(index + (k1, k2)[i%2]) % len(alphabet)];  
 **continue** index = alphabet.find(char)  
 **if** index != -1:  
 result += alphabet[(index + (k1, k2)[i%2]) % len(alphabet)];  
 **continue** result += char  
  
 **return** result  
  
**def** fromDoubleCesar(str, k1, k2):  
 lenalpha = len(string.ascii\_lowercase+string.digits)  
 **return** toDoubleCesar(str, lenalpha - k1 % lenalpha, lenalpha - k2 % lenalpha)  
  
*# задание №4***def** toVigenere(str, keyword):alphabet = string.ascii\_uppercase+string.ascii\_lowercase  
 result = **""****if** any(alphabet.find(char) == -1 **for** char **in** keyword):  
 **raise** ValueError(**"В ключе содержаться недопустимые символы"**)  
  
  
 **for** i **in** range(len(str)):  
 char = str[i]  
 keyIndex = alphabet.find(keyword[i%len(keyword)])  
 charIndex = alphabet.find(char)  
 **if** charIndex != -1:  
 result += alphabet[(charIndex + keyIndex) % len(alphabet)];  
 **continue** result += char  
  
 **return** result  
  
**def** fromVigenere(str, keyword):  
 alphabet = string.ascii\_uppercase + string.ascii\_lowercase  
 result = **""  
  
 if** any(alphabet.find(char) == -1 **for** char **in** keyword):  
 **raise** ValueError(**"В ключе содержаться недопустимые символы"**)  
  
 **for** i **in** range(len(str)):  
 char = str[i]  
 keyIndex = alphabet.find(keyword[i % len(keyword)])  
 charIndex = alphabet.find(char)  
 **if** charIndex != -1:  
 result += alphabet[(charIndex - keyIndex+len(alphabet)) % len(alphabet)];  
 **continue** result += char  
  
 **return** result  
  
  
**def** main():  
 print(**"Задание №1"**)  
 print(toCesar(**"Hello world!Кодовое слово!АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ, абвгдеёжзийклмнопрстуфхцчшщъыьэюя?!"**, 1))  
 print(fromCesar(**"Hello world!Лпепгпё тмпгп!БВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯА, бвгдеёжзийклмнопрстуфхцчшщъыьэюяа?!"**, 1))  
 print(toCesar(**"Hello world!Кодовое слово!АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ, абвгдеёжзийклмнопрстуфхцчшщъыьэюя?!"**, 49))  
 print(fromCesar(**"Hello world!Ъюуюсюф быюсю!ПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЁЖЗИЙКЛМНО, прстуфхцчшщъыьэюяабвгдеёжзийклмно?!"**, 49))  
 print()

print(**"Задание №2"**)  
 print(toCode(**"Hello world "**.upper() + string.ascii\_uppercase, **"Hello"**))  
 print(fromCode(**"DAJJN WNRJO HELOABCDFGIJKMNPQRSTUVWXYZ"**, **"Hello"**))  
 print()  
  
 print(**"Задание №3"**)  
 print(toDoubleCesar(**"Hello world!"**+string.ascii\_lowercase+string.digits+**"?!"**, 1,3))  
 print(fromDoubleCesar(**"Hhmop xrsoe!bedgfihkjmlonqpsrutwvyx0z21436587a9c?!"**, 1,3))  
 print(toDoubleCesar(**"Hello world!"**+string.ascii\_lowercase+string.digits+**"?!"**, 49,27))  
 print(fromDoubleCesar(**"H5yc1 9f4cq!n2p4r6t8vaxcze1g3i5k7m9obqdsfuhwjyl0?!"**, 49, 27))  
 print()  
  
 print(**"Задание №4"**)  
 print(toVigenere(**"ATTACKATDAWN"**, **"LEMON"**))  
 print(fromVigenere(**"LXfOPVEfRNhR"**, **"LEMON"**))  
 print(toVigenere(**"ATTACK AT DAWN!"**,**"LEMON"**))  
 print(fromVigenere(**"LXfOPV Mh OEib!"**,**"LEMON"**))  
 **try**:  
 print(toVigenere(**"ATTACK AT DAWN!"**,**"ЛИМОН"**))  
 **except** ValueError **as** error:  
 print(error)  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()

Результат выполнения программы

**Задание №1**

Hello world!Лпепгпё тмпгп!БВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯА, бвгдеёжзийклмнопрстуфхцчшщъыьэюяа?!

Hello world!Кодовое слово!АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ, абвгдеёжзийклмнопрстуфхцчшщъыьэюя?!

Hello world!Ъюуюсюф быюсю!ПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЁЖЗИЙКЛМНО, прстуфхцчшщъыьэюяабвгдеёжзийклмно?!

Hello world!Кодовое слово!АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ, абвгдеёжзийклмнопрстуфхцчшщъыьэюя?!

**Задание №2**

DAJJN WNRJO HELOABCDFGIJKMNPQRSTUVWXYZ

HELLO WORLD ABCDEFGHIJKLMNOPQRSTUVWXYZ

**Задание №3**

Hhmop xrsoe!bedgfihkjmlonqpsrutwvyx0z21436587a9c?!

Hello world!abcdefghijklmnopqrstuvwxyz0123456789?!

H5yc1 9f4cq!n2p4r6t8vaxcze1g3i5k7m9obqdsfuhwjyl0?!

Hello world!abcdefghijklmnopqrstuvwxyz0123456789?!

**Задание №4**

LXfOPVEfRNhR

ATTACKATDAWN

LXfOPV Mh OEib!

ATTACK AT DAWN!

В ключе содержаться недопустимые символы

Вывод

В ходе выполнения практической работы были получены навыки работы со строками в языке Python, а также опыт применения алгоритмов шифрования строк.