**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Российский Университет Транспорта (МИИТ)»**

**Институт экономики и финансов**

**Кафедра «Информационные системы цифровой экономики»**

**О Т Ч Е Т**

**по дисциплине: «Информатика»**

**о выполнении практических работ**

**на тему «Программирование в Python»**

**Выполнила: студентка ЭББд-111**

**Иванова И.И.**

.

**Проверила**: **ст.преп. Дмитриева Т.М.**

**Москва 2019 г.**

**Содержание**

[Возможности модуля math 3](#_Toc21372358)-6

[Строки 7](#_Toc21372348)-11

[Списки 12](#_Toc21372349)-18

[Матрицы 19-27](#_Toc21372352)

[Кортежи и словари 28-31](#_Toc21372355)

[Графика 32-33](#_Toc21372357)

[Типовые алгоритмы 34-36](#_Toc21372358)

**Возможности модуля math.**

Пример 1.

Постановка задачи:

Заданное числовое значение округлить до ближайшего большего по значению числа.

Текст программы:

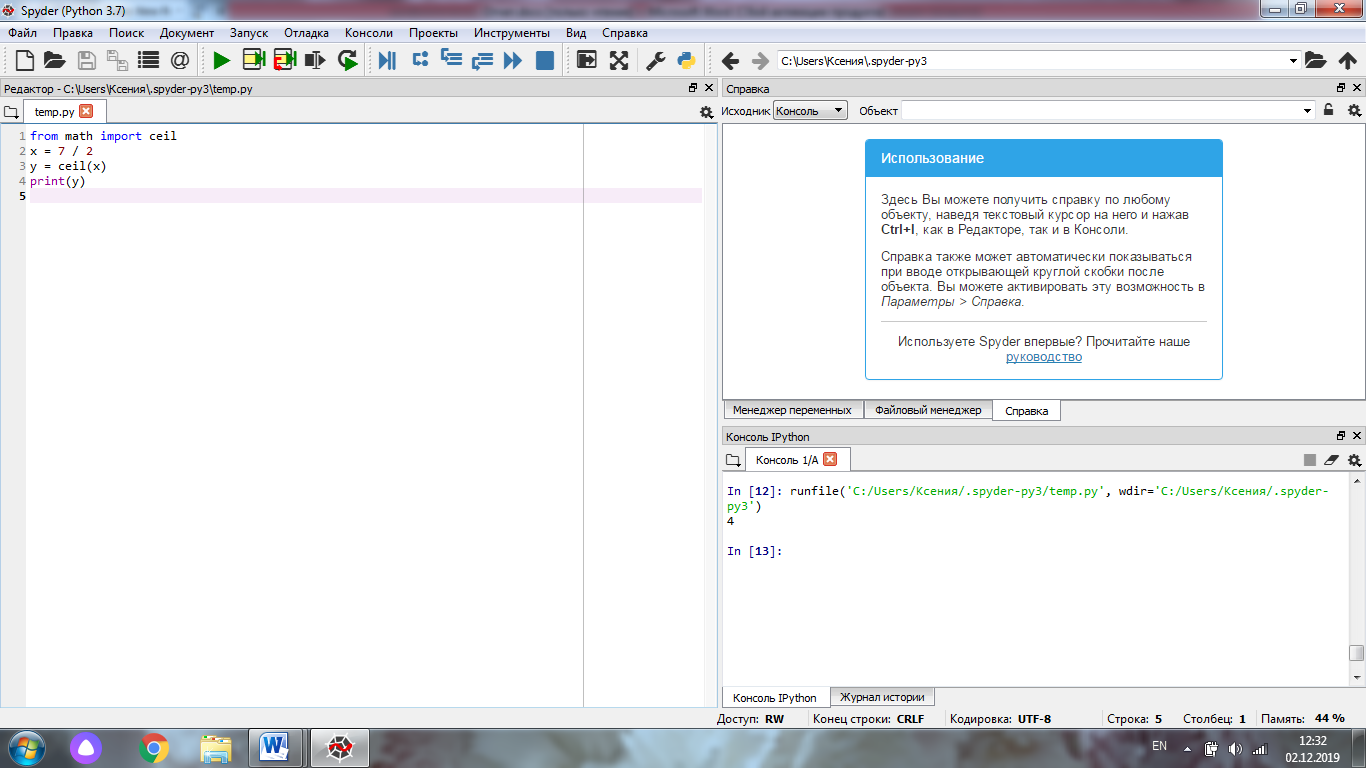
from math import ceil

x = 7 / 2

y = ceil(x)

print(y)

Результаты выполнения:



Пример 2.

Постановка задачи:

К заданным числовым значениям применить функции по выполнению арифметических действий с помощью модуля math.

Текст программы:

import math

n1 = math.pow(2, 3) # возведение числа 2 в степень 3

print(n1)

n2 = 2\*\*3 # ту же самую операцию можно выполнить так

print(n2)

print(math.sqrt(9)) # возведение в квадрат

print(math.ceil(4.56)) # ближайшее наибольшее целое число

print(math.floor(4.56)) # ближайшее наименьшее целое число

print(math.degrees(3.14159)) # перевод из радиан в градусы

print(math.radians(180)) # перевод из градусов в радианы

print(math.cos(math.radians(60))) # косинус

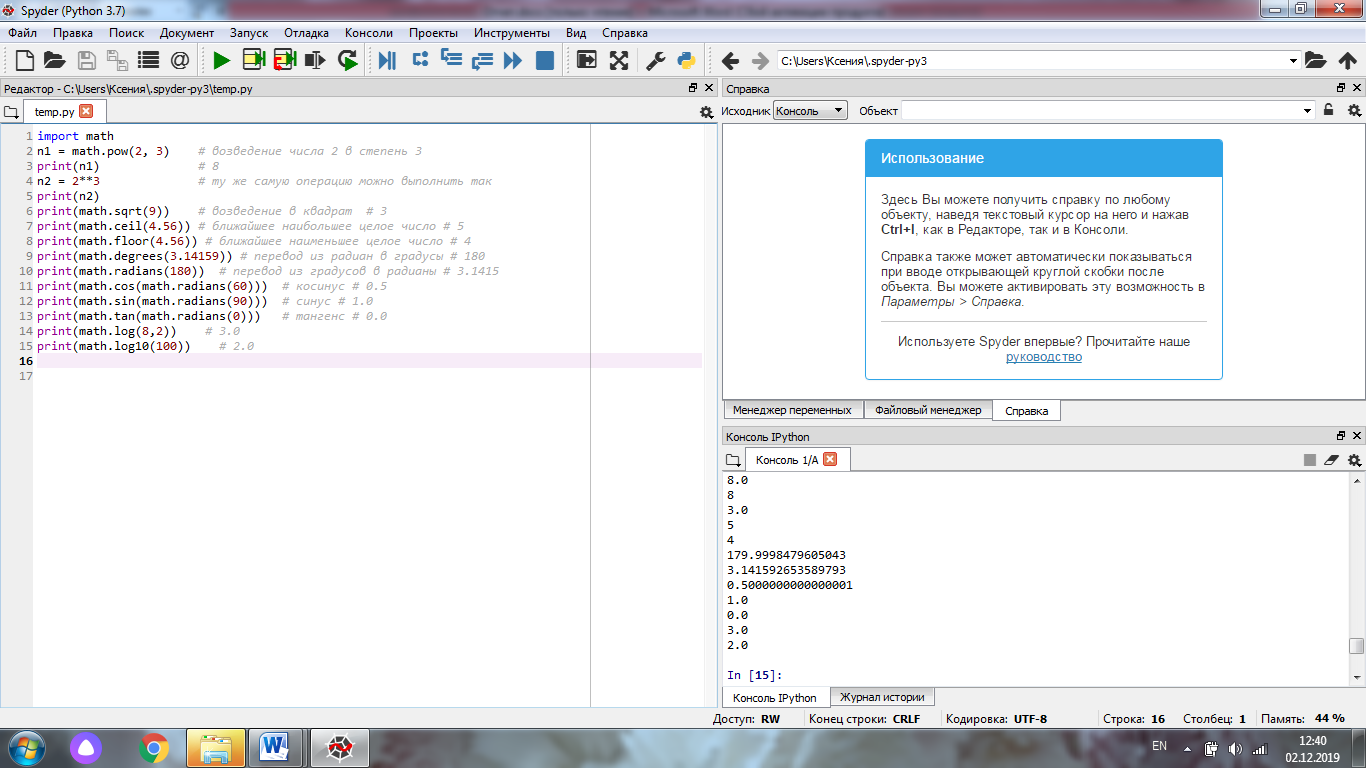
print(math.sin(math.radians(90))) # cинус

print(math.tan(math.radians(0))) # тангенс

print(math.log(8,2))

print(math.log10(100))

Результаты выполнения:



Пример 3.

Постановка задачи:

К заданным числовым значениям применить функции по выполнению арифметических действий с помощью модуля math.

Текст программы:

import math

print(math.sin(math.pi/4))

print (math.cos(math.pi))

print(math.tan(math.pi/6))

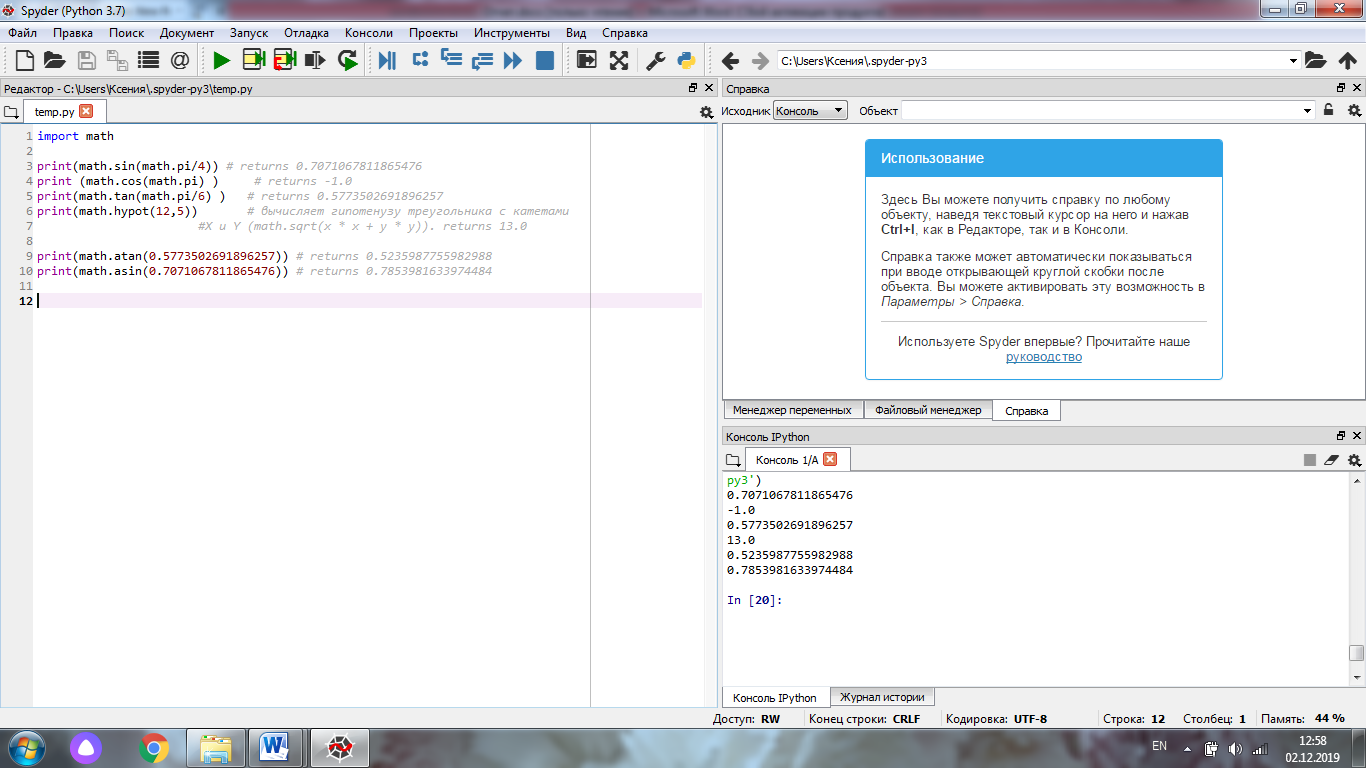
print(math.hypot(12,5)) # вычисляет гипотенузу треугольника с катетами X и

Y (math.sqrt(x\*x + y\*y)). returns 13.0

print(math.atan(0.5773502691896257))

print(math.asin(0.7071067811865476))

Результаты выполнения:



Пример 4.

Постановка задачи:

Напишите программу для решения примера (по вариантам). Предусмотрите проверку деления на ноль. Все необходимые переменные пользователь вводит через консоль. Запись |пример| означает «взять по модулю», т.е. если значение получится отрицательным, необходимо сменить знак с минуса на плюс.

|1 - a\*b^c - a\*(b^2-c^2) + (b-c+a)\*(12+b)/(c-a)|

Текст программы:

import math

a= int(input("Введите число:"))

b= int(input("Введите число:"))

c= int(input("Введите число:"))

x= a\*b\*\*c

y= a\*(b\*\*2-c\*\*2)

z=(12+b)\*(b-c+a)

w=c-a

r=z/w

e=1-x-y+r

if w==0:

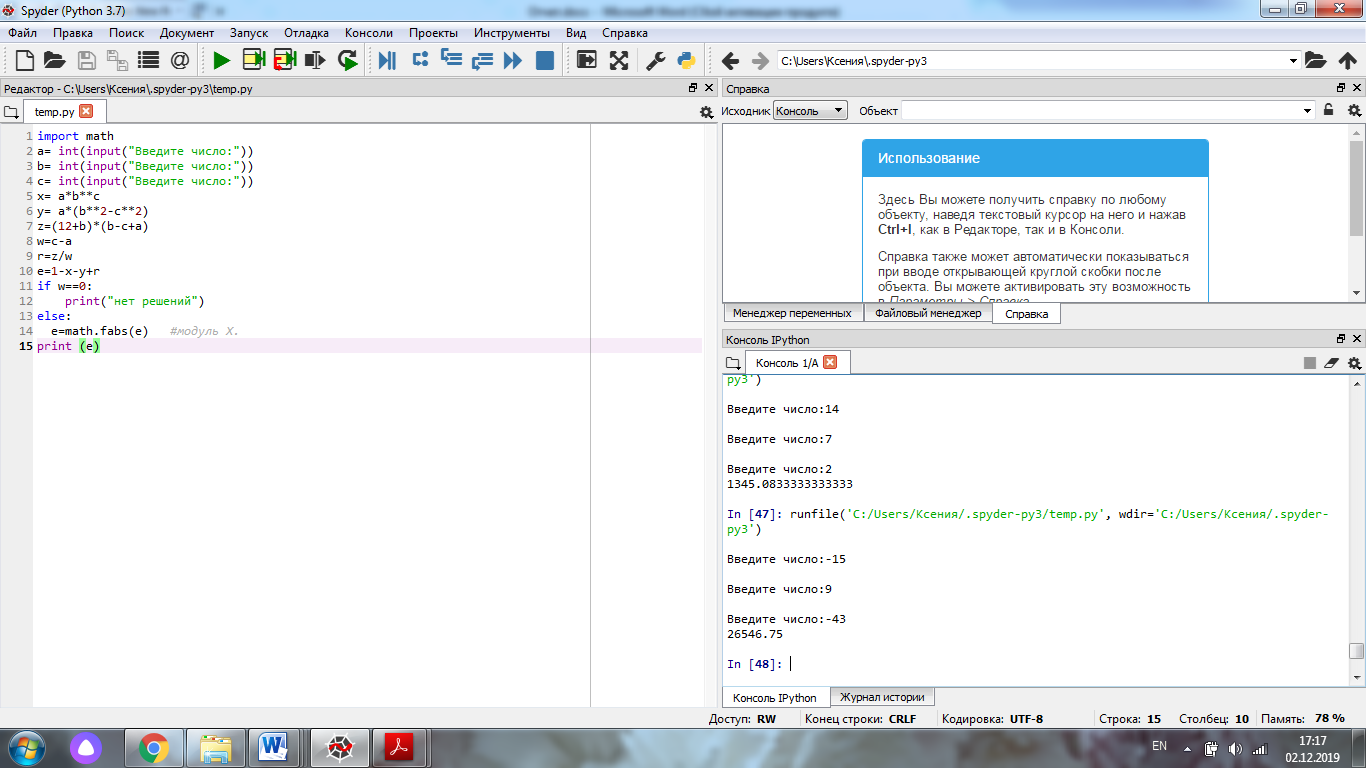
print("нет решений")

else:

e=math.fabs(e) #модуль X.

print (e)

Результаты выполнения:



Блок-схема

1

Начало

Ввод a,b,c

x= a\*b\*\*c

y= a\*(b\*\*2-c\*\*2)

z=(12+b)\*(b-c+a)

w=c-a

r=z/w

e=1-x-y+r

W=0

Нет решений

Вывод e

e=math.fabs(e)

Конец

2

3

да

нет

4

5

6

**Строки.**

Пример 1.

Постановка задачи:

Задать строки тремя разными способами, используя одинарные, двойные и тройные скобки. Результат вывести на экран.

Текст программы:

my\_string = "Добро пожаловать в Python!"

print(my\_string)

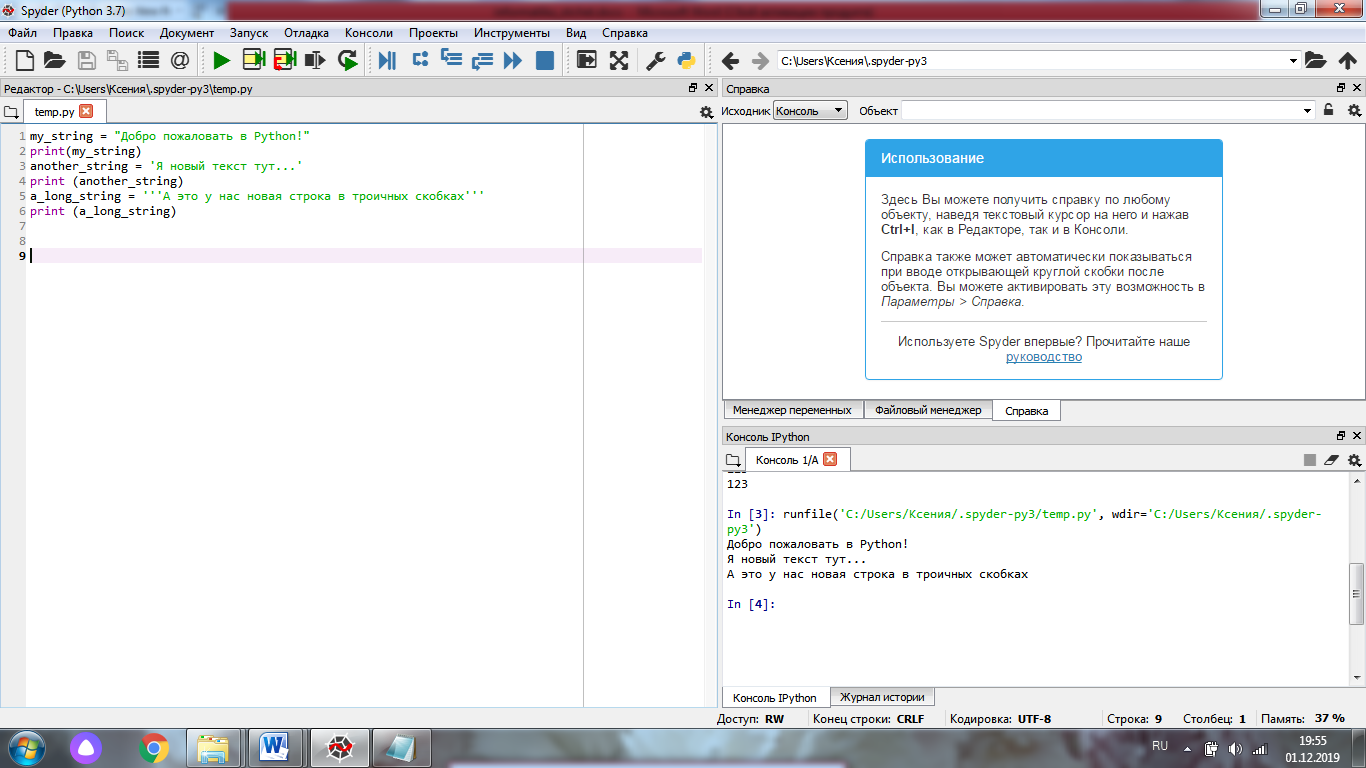
another\_string = 'Я новый текст тут...'

print (another\_string)

a\_long\_string = '''А это у нас новая строка в троичных скобках'''

print (a\_long\_string)

Результаты выполнения:



Пример 2.

Постановка задачи:

Ещё один способ создания строки, при помощи метода str.

Текст программы:

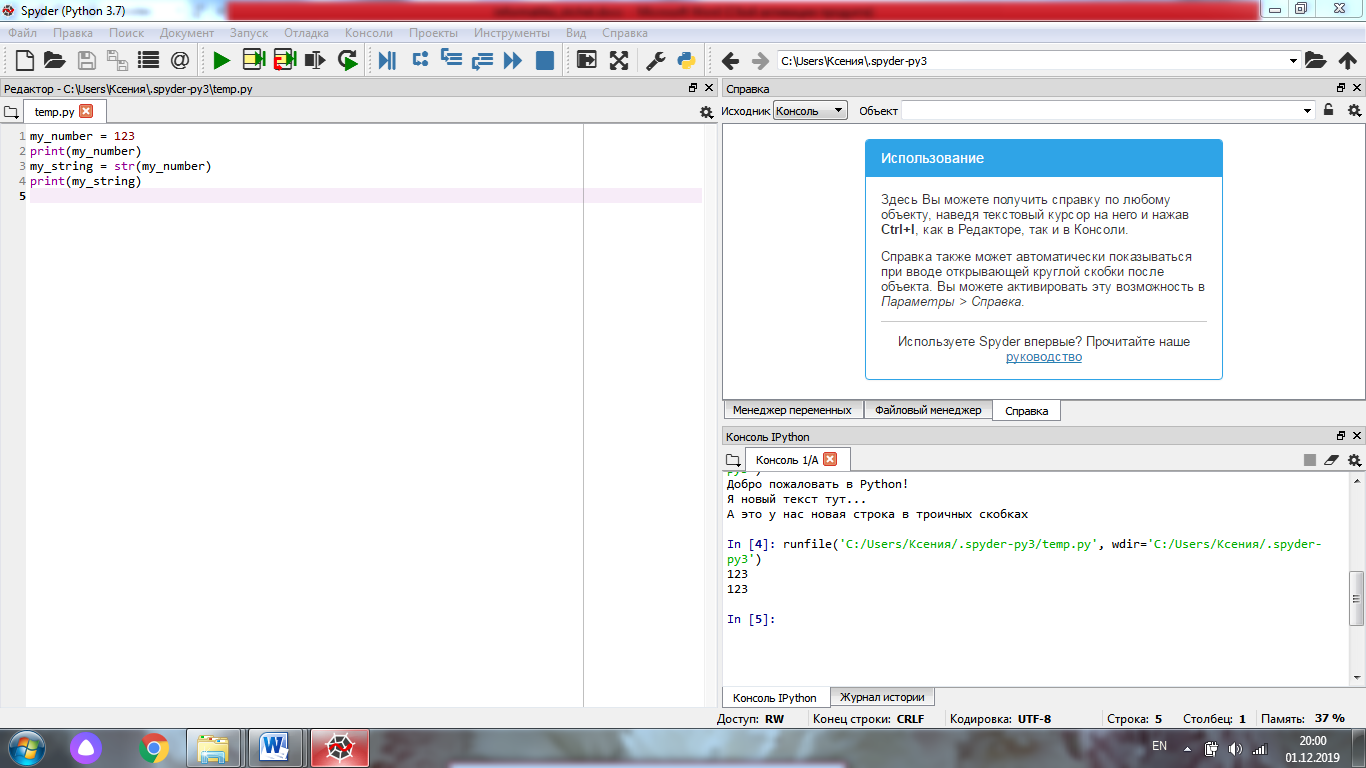
my\_number = 123

print(my\_number)

my\_string = str(my\_number)

print(my\_string)

Результаты выполнения:



Пример 3.

Постановка задачи:

Задана строка. Перевести ее в верхний регистр методом upper(), а потом - в нижний регистр методом lower(). Результаты вывести на экран.

Текст программы:

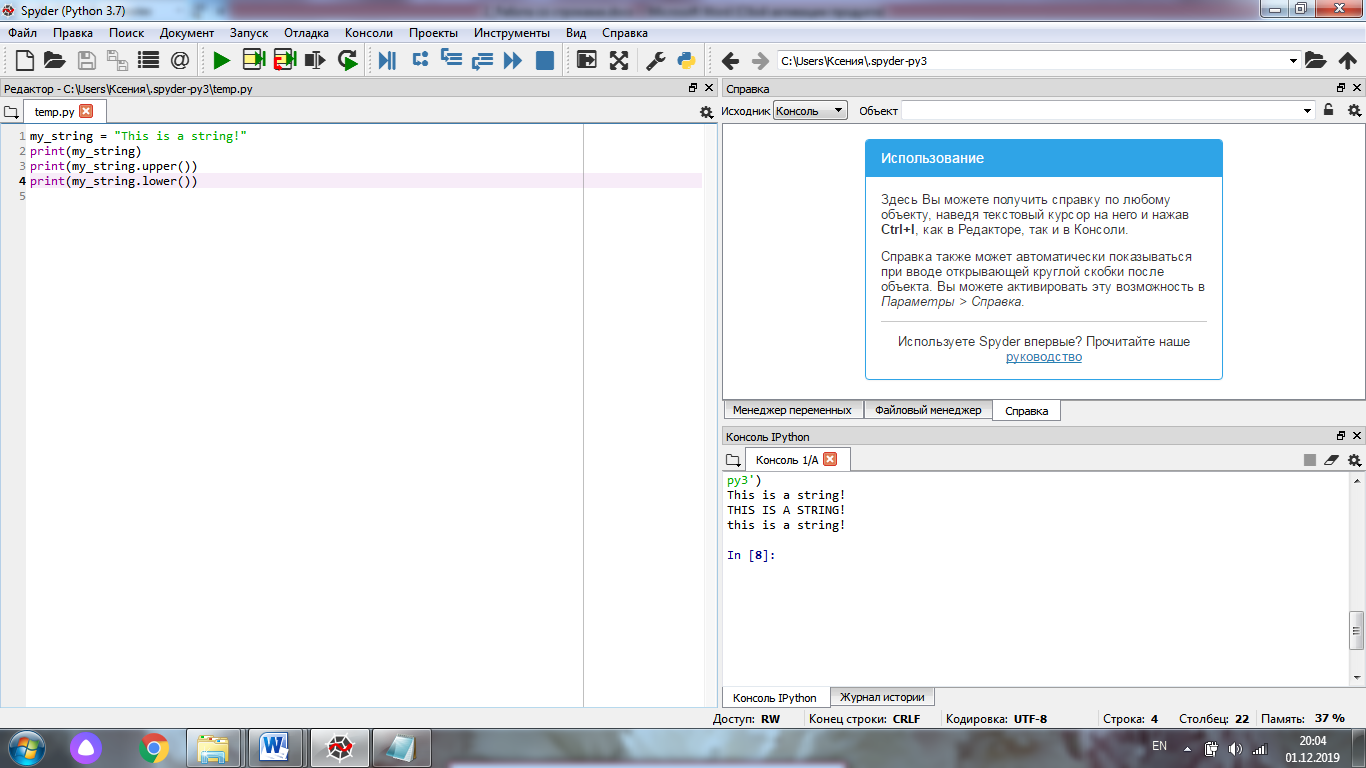
my\_string = "This is a string!"

print(my\_string)

print(my\_string.upper())

print(my\_string.lower())

Результаты выполнения:



Пример 4.

Постановка задачи:

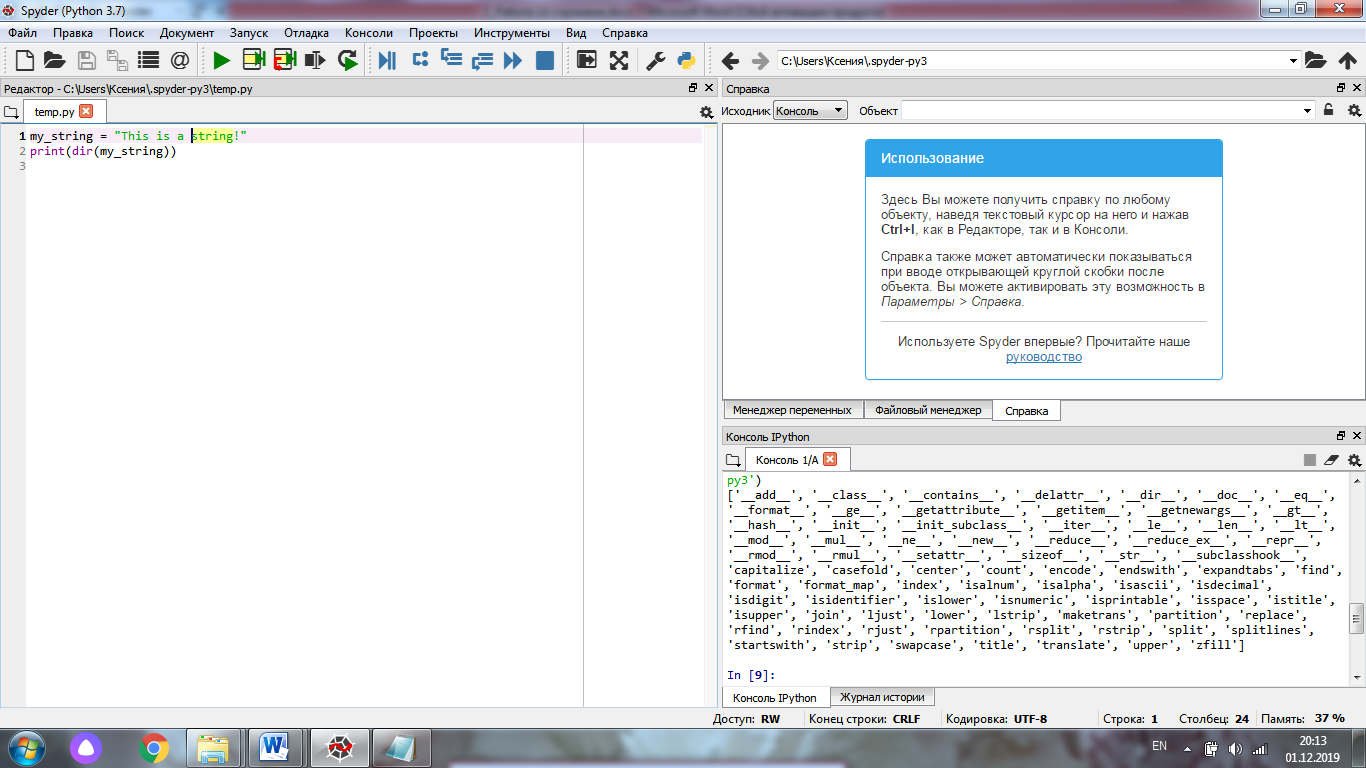
Задана строка. Получить на экране список всех методов, применимых к строкам:

Текст программы:

my\_string = "This is a string!"

print(dir(my\_string))

Результаты выполнения:



Пример 5.

Постановка задачи:

Заданы две строки. Получить срезы строк. Результаты вывести в консоль.

Текст программы:

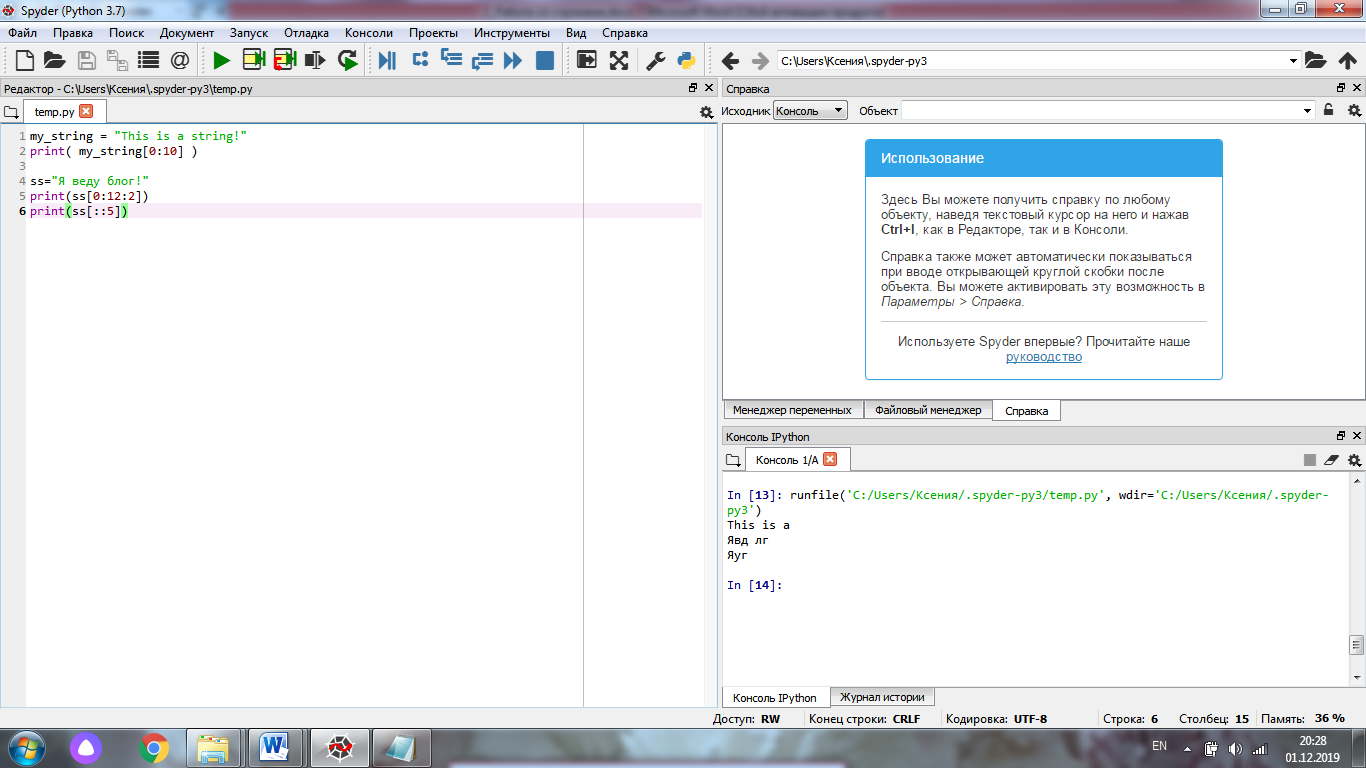
my\_string = "This is a string!"

print( my\_string[0] ) # T

ss=”Я веду блог!”

print(ss[0:12:2])  
print(ss[::5])

Результаты выполнения:



Пример 6.

Постановка задачи:

Задана строка символов. Вычислить католичество символов в строке и подсчитать, сколько раз в строке повторяется тот или иной символ. Результат отобразить в консоли.

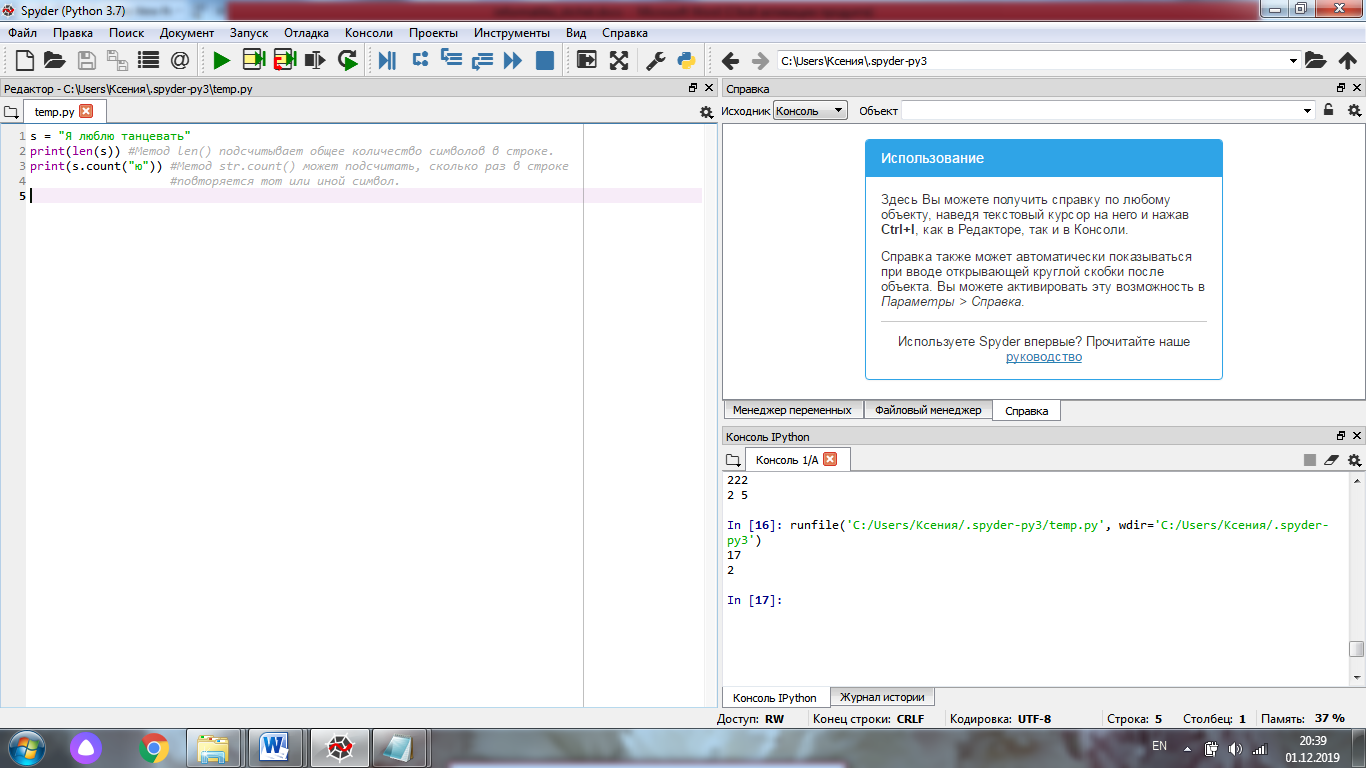
Текст программы:

s= "я люблю танцевать"

print(len(s))

print(s.count("ю"))

Результаты выполнения:



Пример 7.

Постановка задачи:

Ввести строку s, далее подсчитать общее кол-во символов в строке.

Ввести числовое значение - a, затем преобразовать его к строковому типу.

Текст программы:

s = input()

print(len(s))

a = input()

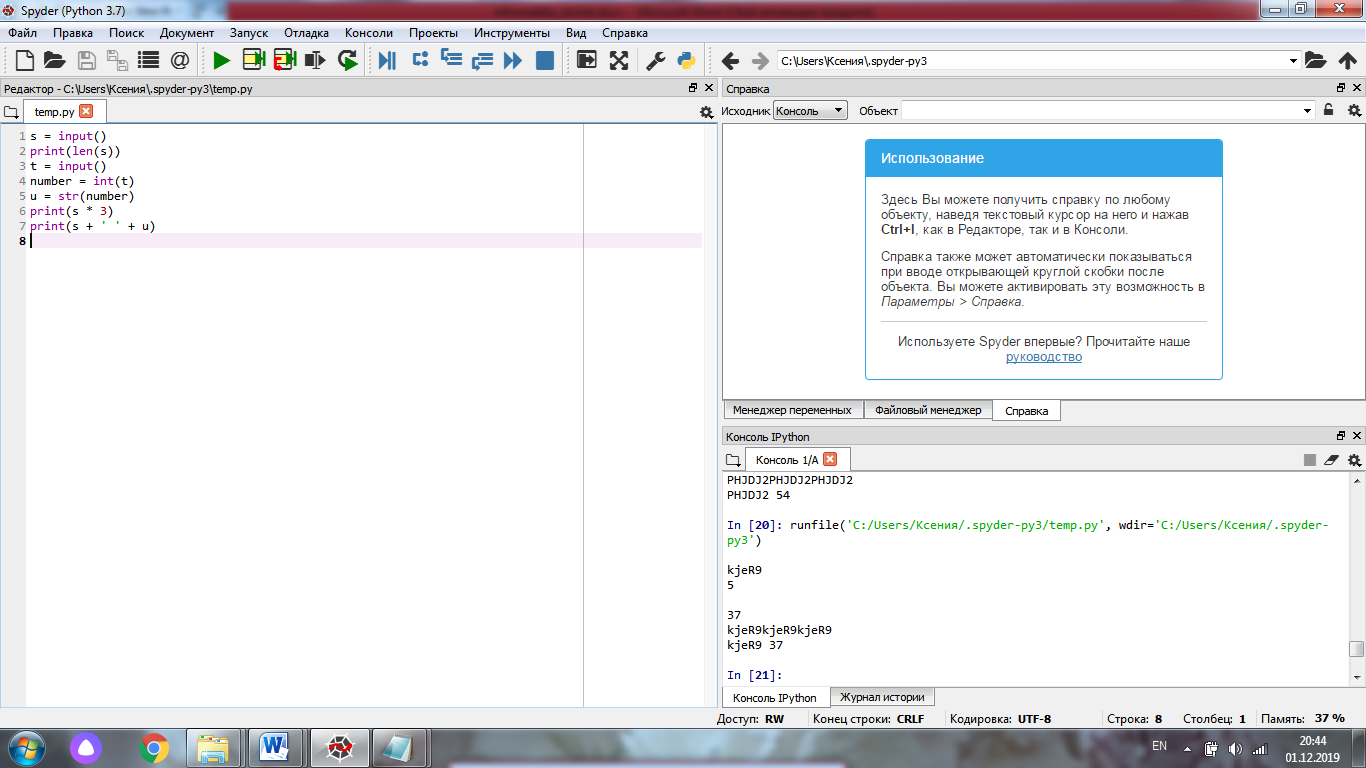
number = int(a)

u = str(number)

print(s \* 3)

print(s + ' ' + u)

Результаты выполнения:



Пример 8.

Постановка задачи:

Дана строка. Определить индекс символа в строке, используя метод str.find(). Результат вывести в консоль.

Текст программы:

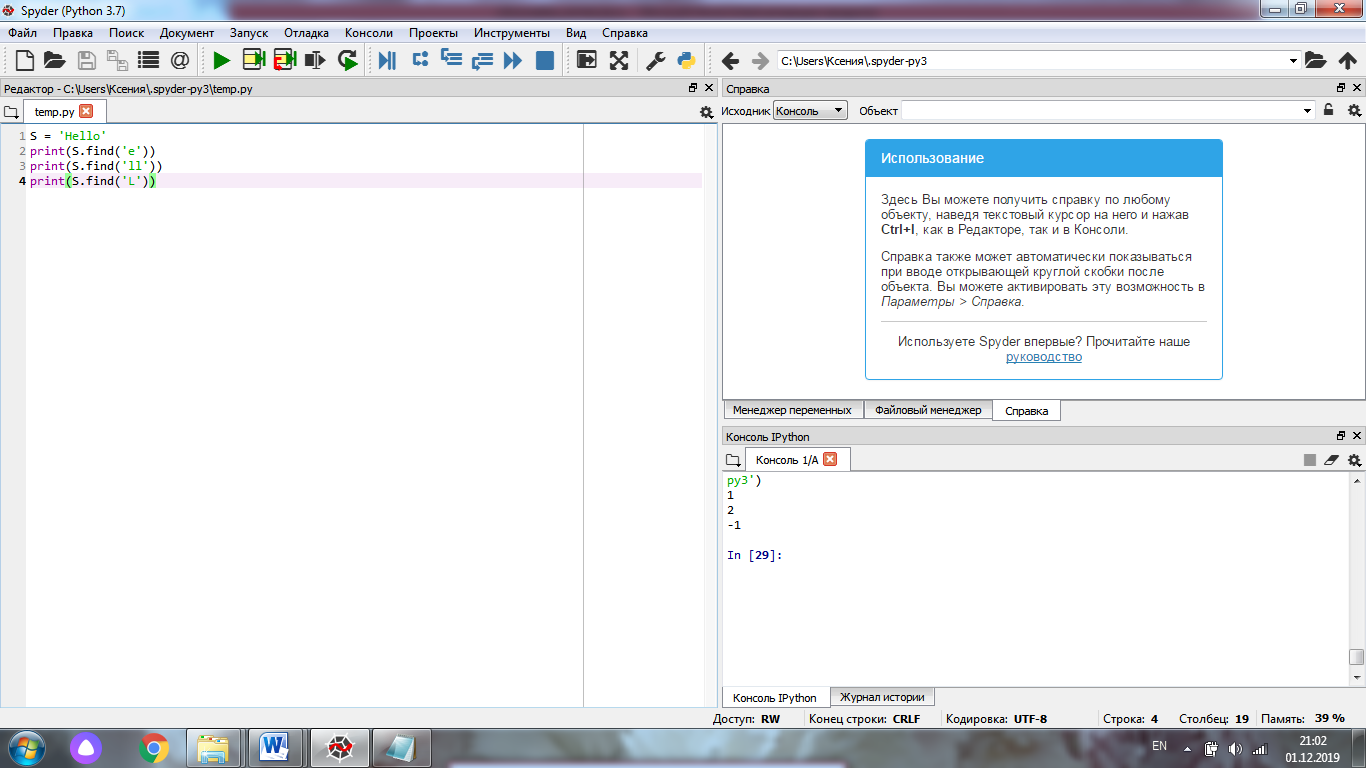
S = 'Hello'

print(S.find('e'))

print(S.find('ll'))

print(S.find('L'))

Результаты выполнения:



**Списки.**

Пример 1.

Постановка задачи:

Задан список числовых значений. Очистить список и вывести результат в консоль. Для заданного списка определить индекс элемента по значению элемента. Результат вывести в консоль.

Текст программы:

a = [8, 7, 6, 5, 4]

print(a)

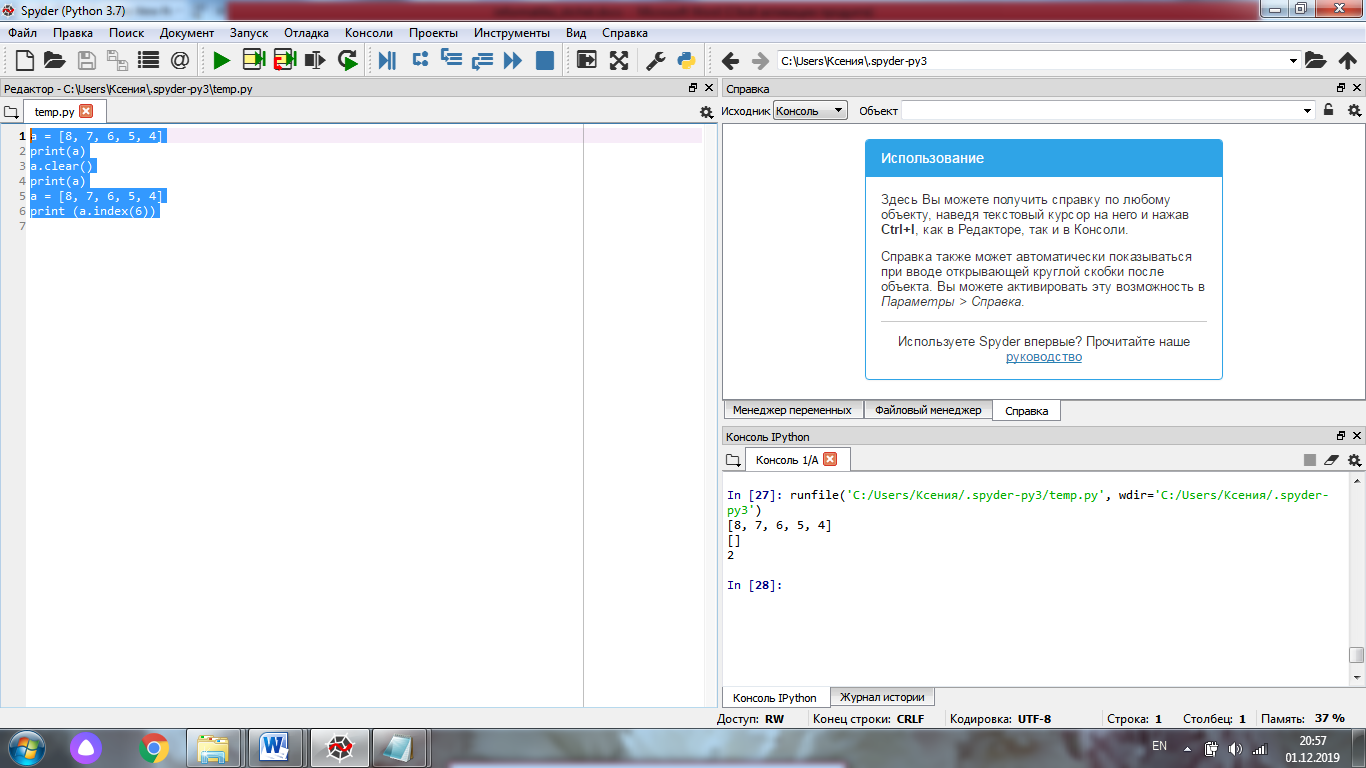
a.clear()

print(a)

a = [8, 7, 6, 5, 4]

print (a.index(6))

Результаты выполнения:



Пример 2.

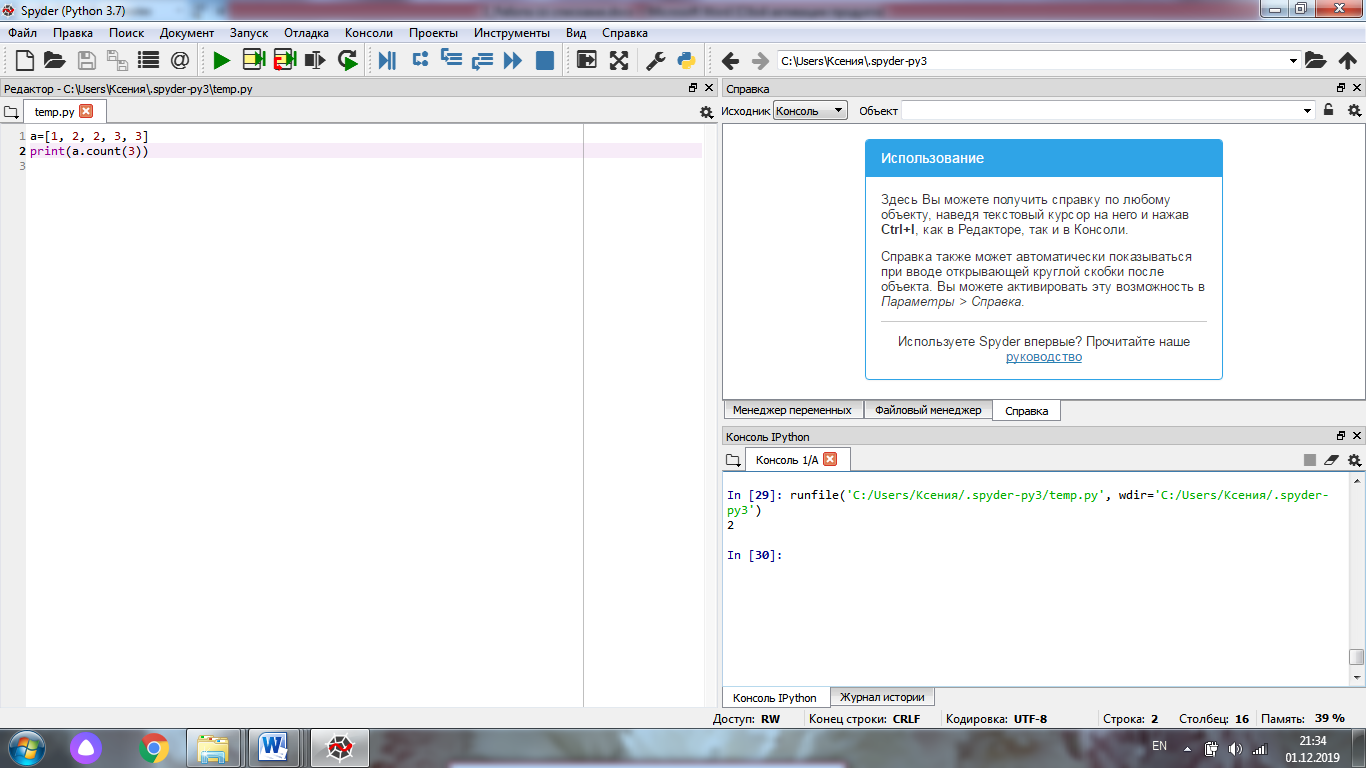
Постановка задачи:

Возвращение количества вхождений элементов в список:

Текст программы:

a=[1, 2, 2, 3, 3]

print(a.count(3))



Пример 3.

Постановка задачи:

Дан список числовых значений. Применить заданные методы для работы со списками. Результаты вывести в консоль.

Текст программы:

a = [1, 2]

b=[8, 9, 2]

a.insert(0, 5) #Вставка элемента в позицию i.

a.append(7) #Добавление элемента в список

print(a)

a.extend(b) #Расширение существующего списка добавлением из другого списка

print(a)

c=a+b #Объединение списков

print(c)

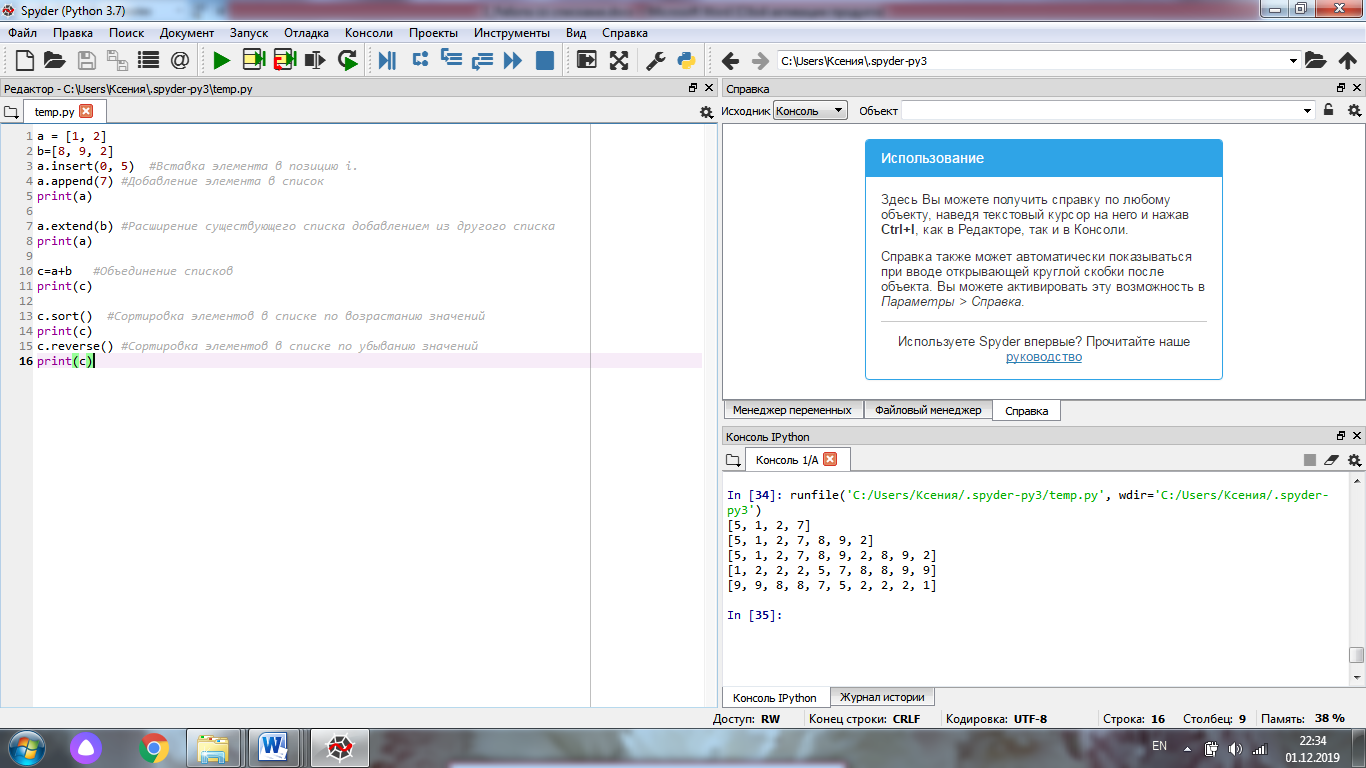
c.sort() #Сортировка элементов в списке по возрастанию значений

print(c)

c.reverse() #Сортировка элементов в списке по убыванию значений

print(c)

Результаты выполнения:



Пример 4.

Постановка задачи:

Задан список числовых значений. Удалить элемент списка из позиции i. Результат вывести вконсоль.

Текст программы:

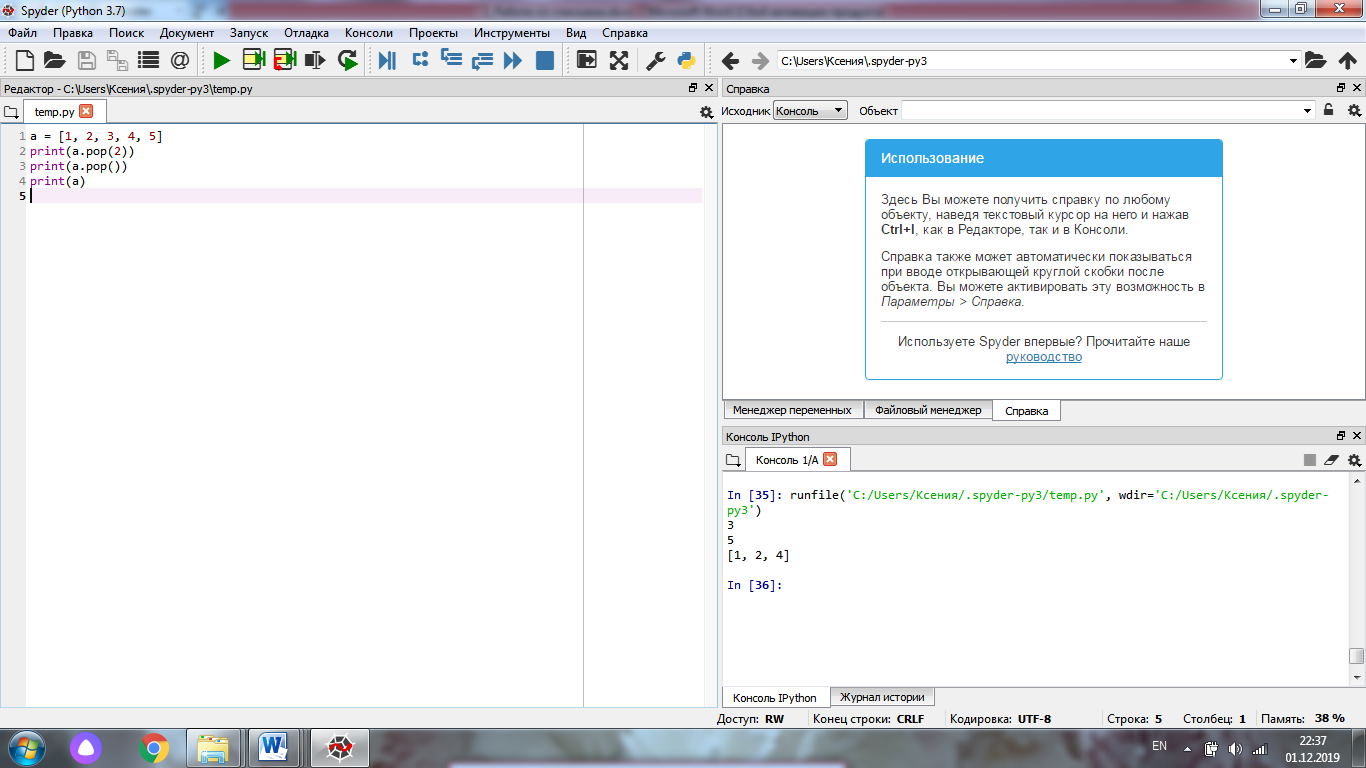
a = [1, 2, 3, 4, 5]

print(a.pop(2))

print(a.pop())

print(a)

Результаты выполнения:



Пример 5.

Постановка задачи:

Задан список. Выполнить вывод всех элементов списка больше предыдущего элемента:

Текст программы:

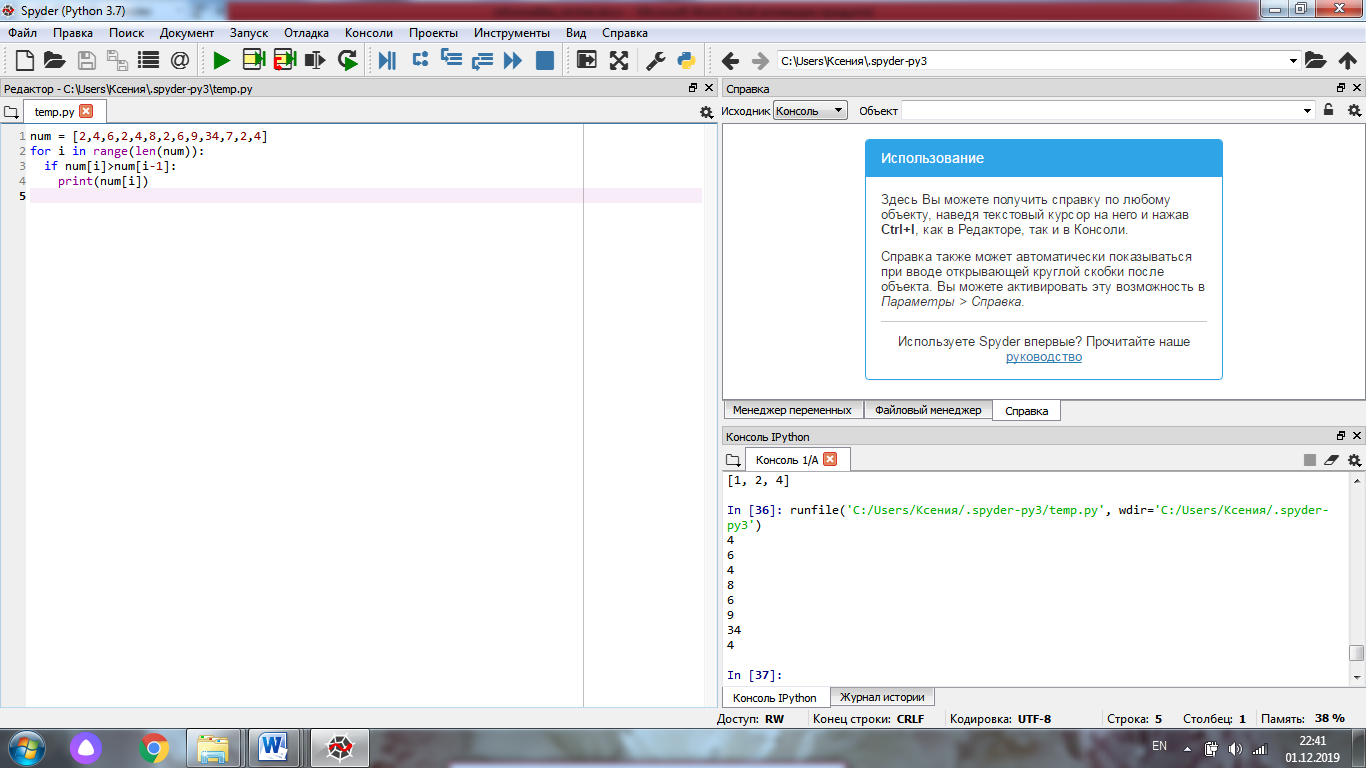
num = [2,4,6,2,4,8,2,6,9,34,7,2,4]

for i in range(len(num)):

if num[i]>num[i-1]:

print(num[i])

Результаты выполнения:



Пример 6.

Постановка задачи:

Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящий на противоположном конце этого списка.

Текст программы:

a = [int(i) for i in input().split()] # метод split разделяет строку по разделителю

if len(a) < 2:

print(' '.join([str(i) for i in a]))

else:

new = []

new.append(a[1] + a[len(a) - 1])

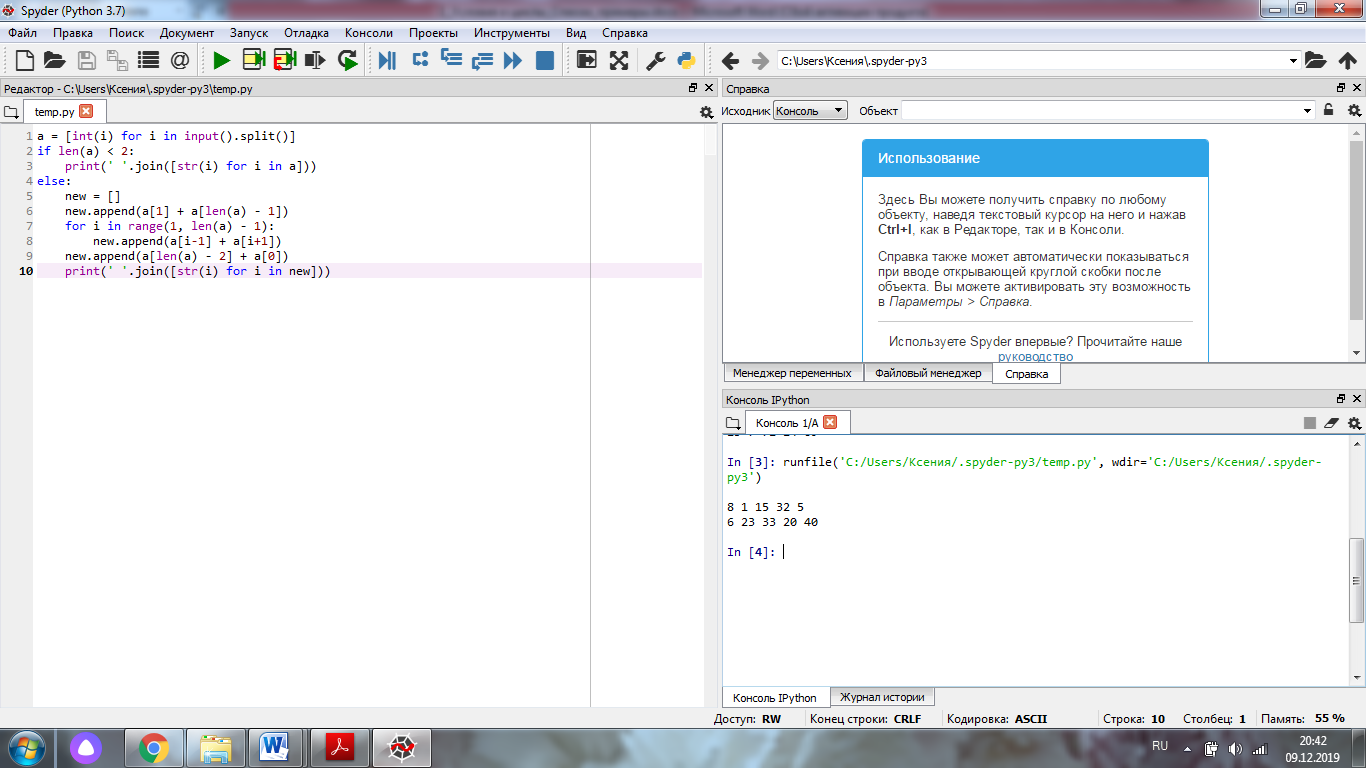
for i in range(1, len(a) - 1):

new.append(a[i-1] + a[i+1])

new.append(a[len(a) - 2] + a[0])

print(' '.join([str(i) for i in new]))

Результаты выполнения:



Блок-схема

Начало

Ввод a

new = []

new.append(a[1] + a[len(a) - 1]

len(a) < 2

for i in range(1, len(a) - 1)

new.append(a[i-1] + a[i+1])

new.append(a[len(a) - 2] + a[0])

''.join([str(i) for i in a]

Конец

да

нет

1

2

3

4

5

6

Пример 7.

Постановка задачи:

Задан список числовых значений. Нужно отсортировать список, но при этом нужно сохранить знание о том, где стоял элемент до сортировки. Для этого можно применить такой трюк: заменим число x на кортеж (x, i), где i — номер элемента в списке. И после этого отсортируем.

Текст программы:

my\_list = [39, 12, 21, 77, 21, 51, 48, 21, 42, 76]

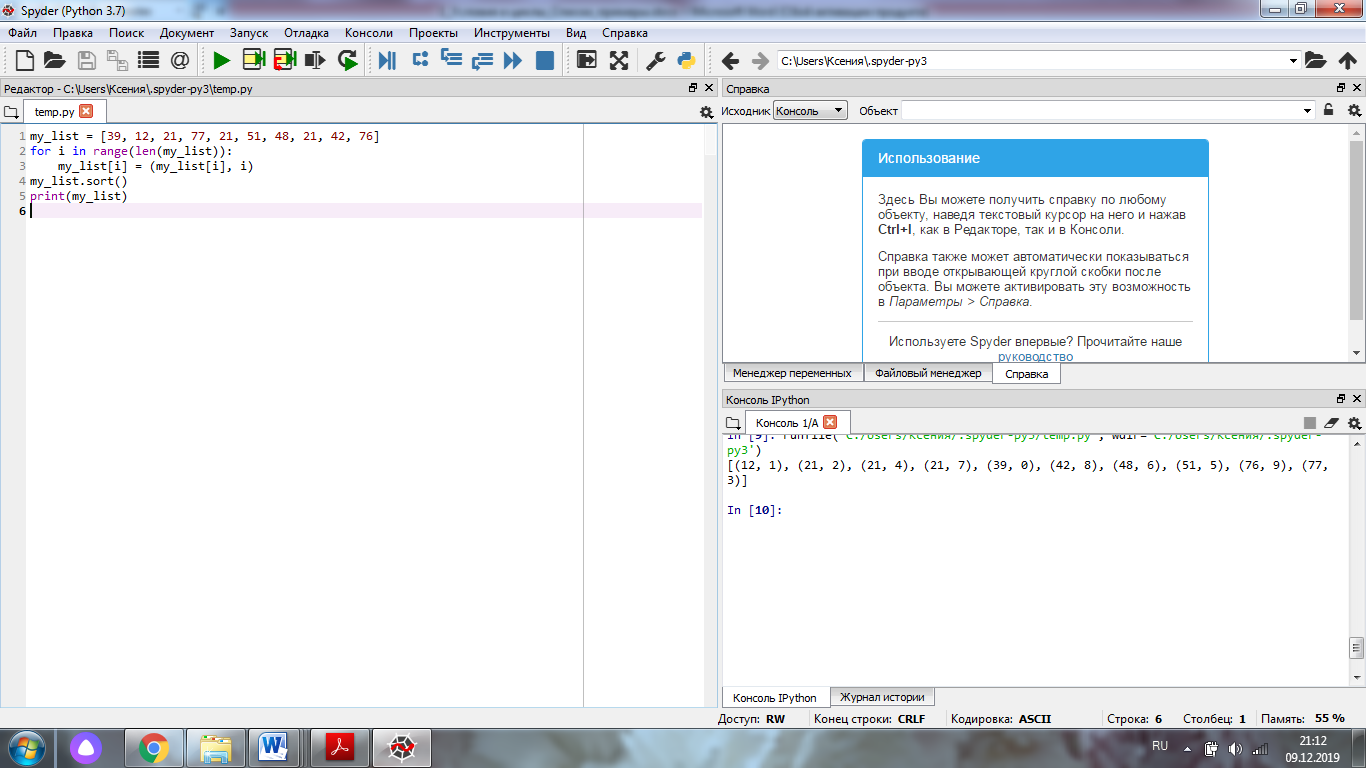
for i in range(len(my\_list)):

my\_list[i] = (my\_list[i], i)

my\_list.sort()

print(my\_list)

Результаты выполнения:



Пример 8.

Постановка задачи:

Задан список. Выполнить поиск максимального (минимального) элемента в списке. Результаты вывести в консоль.

Текст программы:

mas = [3, 5, 67, -65, 34, 21] # задаем список

maximum = mas[0] # предположим, что максимальный элемент равен mas[0]

for i in range(1, len(mas)):

if mas[i] > maximum:

maximum = mas[i]

print(maximum)

minimum = mas[0] # предположим, что минимальный элемент равен mas[0]

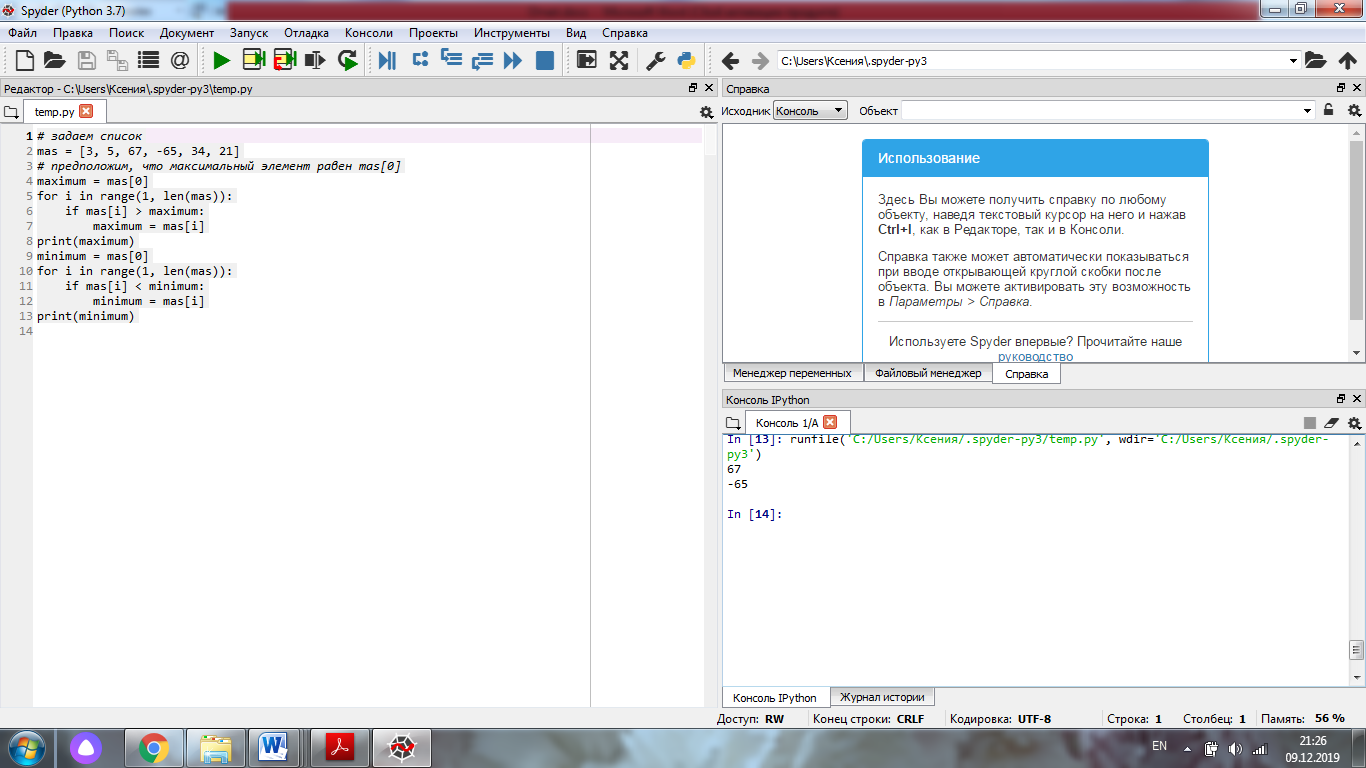
for i in range(1, len(mas)):

if mas[i] < minimum:

minimum = mas[i]

print(minimum)

Результаты выполнения:



Пример 9.

Постановка задачи:

Создать список случайных чисел. Вывести номера элементов списка, которые больше своих соседей.

Текст программы:

import random

mas = [random.randint(0, 100) for i in range(20)] #Создадим список случайных чисел

print(mas) #Выведем его на экран

#Начиная со второго элемента массива mas[1] и до предпоследнего

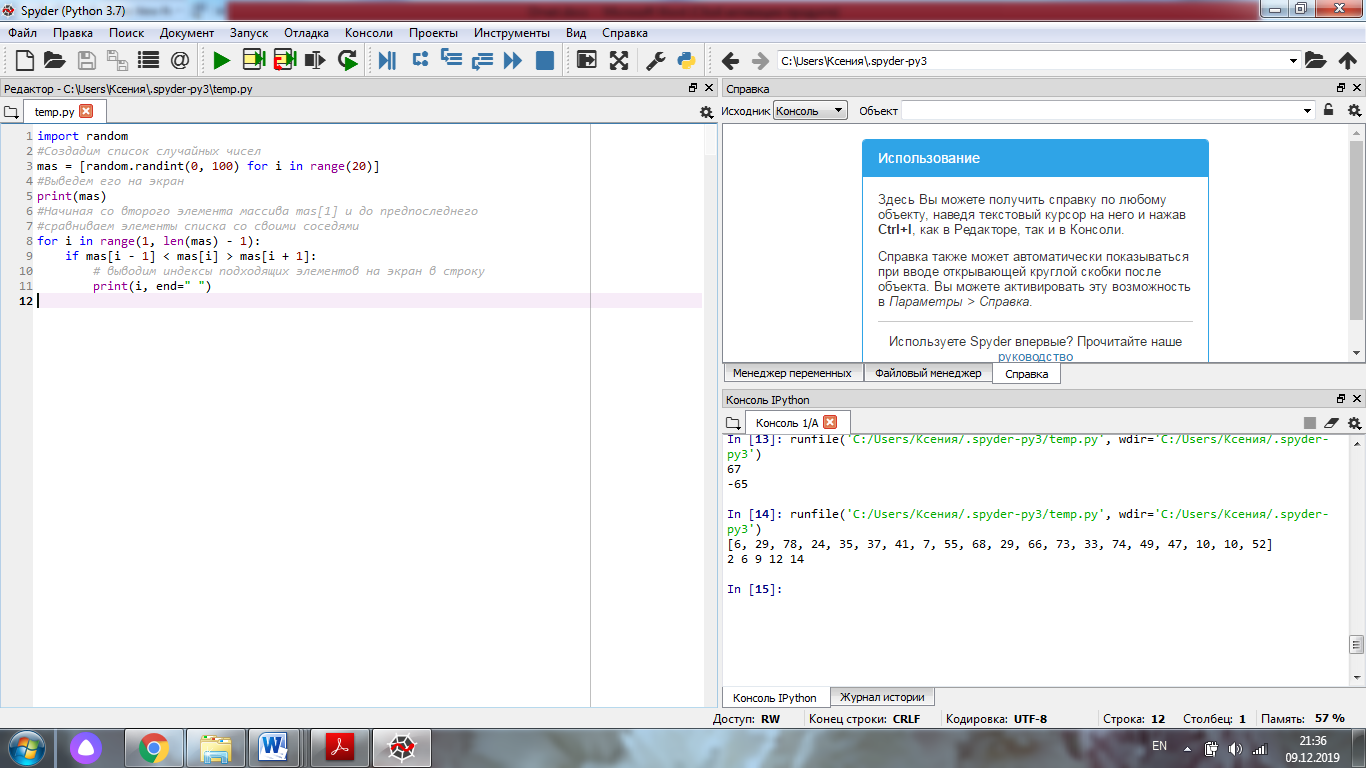
#сравниваем элементы списка со своими соседями

for i in range(1, len(mas) - 1):

if mas[i - 1] < mas[i] > mas[i + 1]:

print(i, end=" ") # выводим индексы подходящих элементов на экран в строку

Результаты выполнения:



Пример 10.

Постановка задачи:

Создать список случайных чисел. Найти произведение элементов списка, кратных 6 и оканчивающихся на 8. Если таких элементов нет - сообщить об этом

Текст программы:

import random

#Создадим список случайных чисел

mas = [random.randint(0, 100) for i in range(20)]

print(mas) #Выведем его на экран

#Так как ищем произведение - присвоим переменной значение 1

p = 1

for i in range(len(mas)):

if mas[i] % 6 == 0 and mas[i] % 10 == 8:

p \*= mas[i]

#Если произведение по-прежнему ноль - значит таких элементов нет

if p == 1:

print("Нет таких элементов")

else:

print(p)

Результаты выполнения:

