Лабораторная работа №3

Тема: Стандартные типы данных, коллекции, функции, модули.

Цель: освоить базовый синтаксис языка Python, приобрести навыки работы со стандартными типами данных, коллекциями, функциями, модулями и закрепить их на примере разработки интерактивных приложений.

Выполнил: ст. гр.253503 Белян Д.Д. , Вариант 3

Задание 1. В соответствии с заданием своего варианта составить программу для вычисления значения функции с помощью разложения функции в степенной ряд. Задать точность вычислений eps.

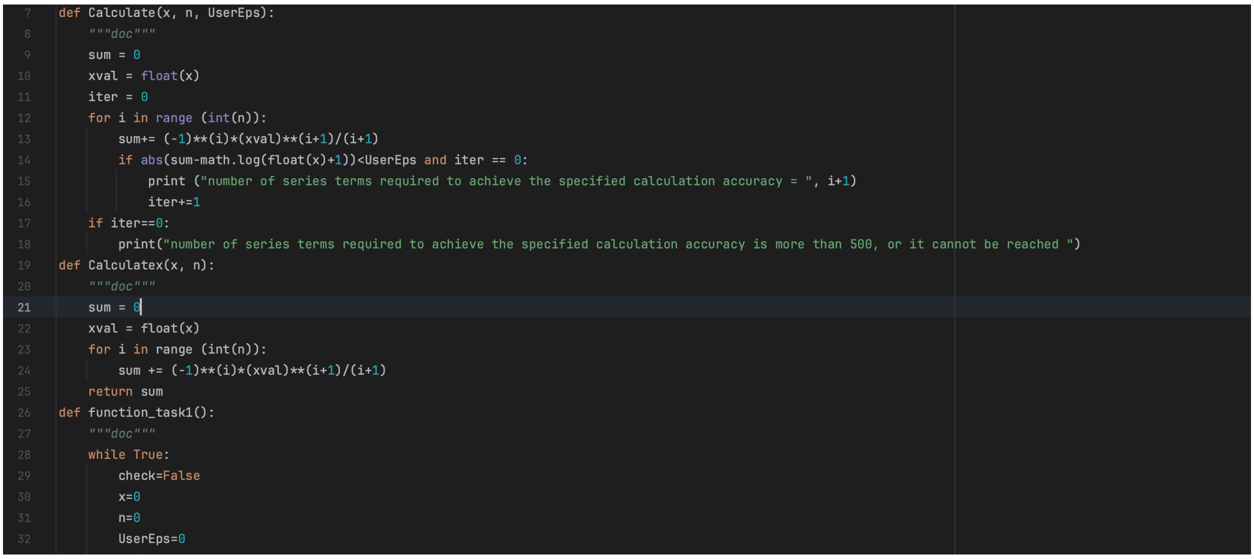
Предусмотреть максимальное количество итераций, равное 500.

Вывести количество членов ряда, необходимых для достижения указанной точности вычислений. Результат получить в виде:

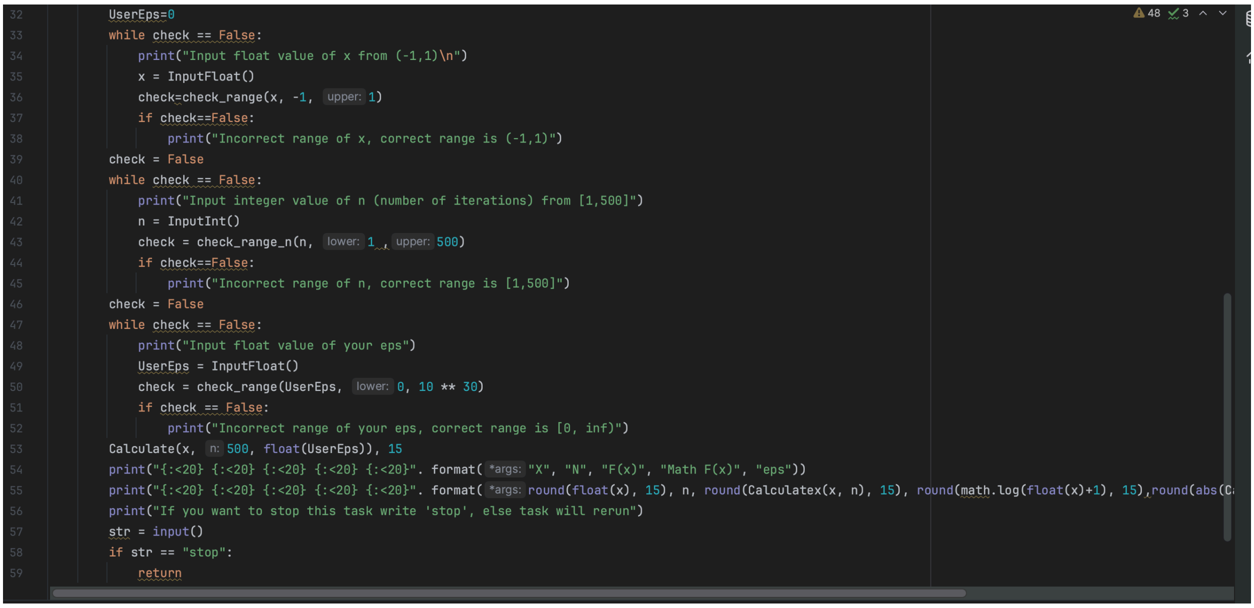
x	n	$F(x)$	$Math\ F(x)$	eps

Здесь x – значение аргумента, F(x) – значение функции, n – количество просуммированных членов ряда, Math F(x) – значение функции, вычисленное с помощью модуля math.

Вар-т	**Условие**
	$\ln(1+x)=\sum\limits_{n=0}^{\infty}(-1)^{n-1}\dfrac{x^n}{n}=x-\dfrac{x^2}{2}+\dfrac{x^3}{3}+\ldots,\ \lvert x\rvert<1$

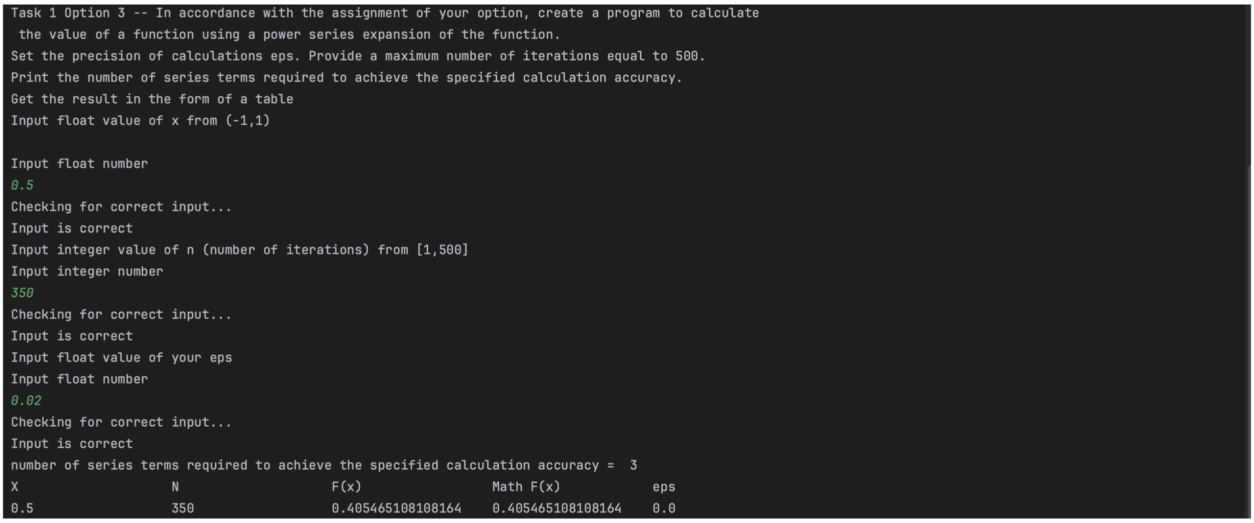


```python
def Calculate(x, n, UserEps):
    """doc"""
    sum = 0
    xval = float(x)
    iter = 0
    for i in range (int(n)):
        sum+= (-1)**(i)*(xval)**(i+1)/(i+1)
        if abs(sum-math.log(float(x)+1))<UserEps and iter == 0:
            print ("number of series terms required to achieve the specified calculation accuracy = ", i+1)
            iter+=1
    if iter==0:
        print("number of series terms required to achieve the specified calculation accuracy is more than 500, or it cannot be reached ")
def Calculatex(x, n):
    """doc"""
    sum = 0
    xval = float(x)
    for i in range (int(n)):
        sum += (-1)**(i)*(xval)**(i+1)/(i+1)
    return sum
def function_task1():
    """doc"""
    while True:
        check=False
        x=0
        n=0
        UserEps=0
```



```
        UserEps=0
        while check == False:
            print("Input float value of x from (-1,1)\n")
            x = InputFloat()
            check=check_range(x, -1,  upper: 1)
            if check==False:
                print("Incorrect range of x, correct range is (-1,1)")
        check = False
        while check == False:
            print("Input integer value of n (number of iterations) from [1,500]")
            n = InputInt()
            check = check_range_n(n,  lower: 1 ,  upper: 500)
            if check==False:
                print("Incorrect range of n, correct range is [1,500]")
        check = False
        while check == False:
            print("Input float value of your eps")
            UserEps = InputFloat()
            check = check_range(UserEps,  lower: 0, 10 ** 30)
            if check == False:
                print("Incorrect range of your eps, correct range is [0, inf)")
        Calculate(x,  n: 500, float(UserEps)), 15
        print("{:<20} {:<20} {:<20} {:<20} {:<20}". format( *args: "X", "N", "F(x)", "Math F(x)", "eps"))
        print("{:<20} {:<20} {:<20} {:<20} {:<20}". format( *args: round(float(x), 15), n, round(Calculatex(x, n), 15), round(math.log(float(x)+1), 15),round(abs(C
        print("If you want to stop this task write 'stop', else task will rerun")
        str = input()
        if str == "stop":
            return
```

Выполнение:

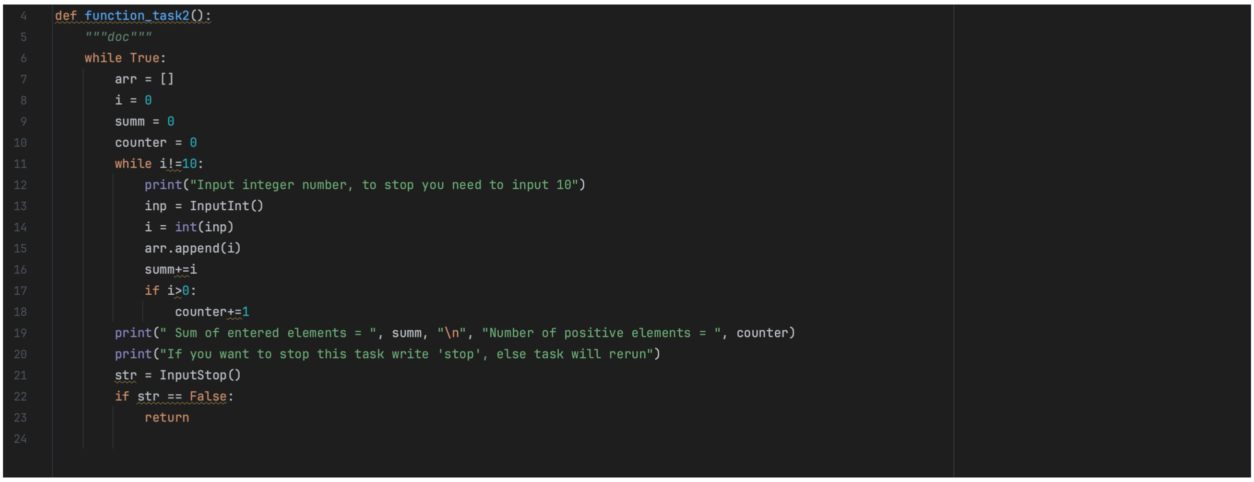


```
Task 1 Option 3 -- In accordance with the assignment of your option, create a program to calculate
 the value of a function using a power series expansion of the function.
Set the precision of calculations eps. Provide a maximum number of iterations equal to 500.
Print the number of series terms required to achieve the specified calculation accuracy.
Get the result in the form of a table
Input float value of x from (-1,1)

Input float number
0.5
Checking for correct input...
Input is correct
Input integer value of n (number of iterations) from [1,500]
Input integer number
350
Checking for correct input...
Input is correct
Input float value of your eps
Input float number
0.02
Checking for correct input...
Input is correct
number of series terms required to achieve the specified calculation accuracy =  3
X                    N                    F(x)                 Math F(x)            eps
0.5                  350                  0.405465108108164    0.405465108108164    0.0
```

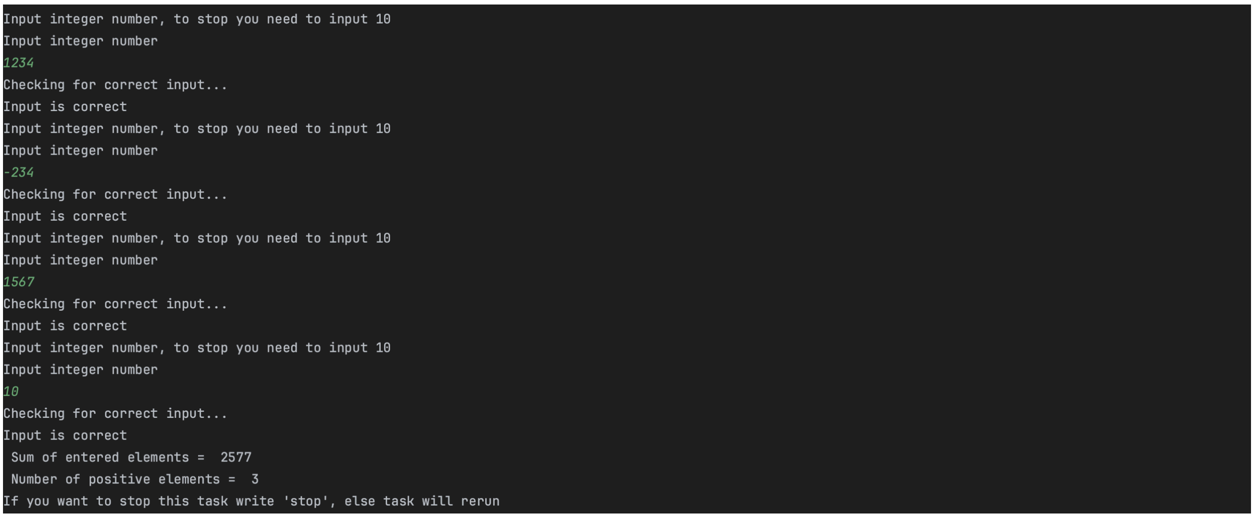
Задание 2. В соответствии с заданием своего варианта составить программу для нахождения суммы последовательности чисел.

Вар-т	Условие
	Организовать цикл, принимающий целые числа и подсчитывающий количество положительных. Окончание – ввод 10



```python
def function_task2():
    """doc"""
    while True:
        arr = []
        i = 0
        summ = 0
        counter = 0
        while i!=10:
            print("Input integer number, to stop you need to input 10")
            inp = InputInt()
            i = int(inp)
            arr.append(i)
            summ+=i
            if i>0:
                counter+=1
        print(" Sum of entered elements = ", summ, "\n", "Number of positive elements = ", counter)
        print("If you want to stop this task write 'stop', else task will rerun")
        str = InputStop()
        if str == False:
            return
```

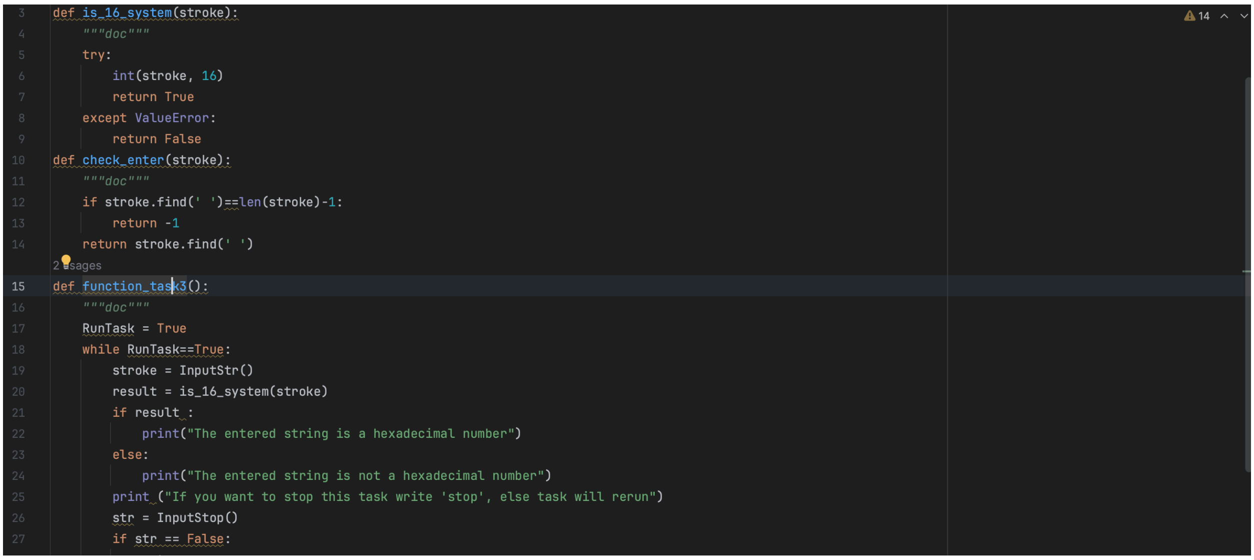
Выполнение:



```
Input integer number, to stop you need to input 10
Input integer number
1234
Checking for correct input...
Input is correct
Input integer number, to stop you need to input 10
Input integer number
-234
Checking for correct input...
Input is correct
Input integer number, to stop you need to input 10
Input integer number
1567
Checking for correct input...
Input is correct
Input integer number, to stop you need to input 10
Input integer number
10
Checking for correct input...
Input is correct
 Sum of entered elements =  2577
 Number of positive elements =  3
If you want to stop this task write 'stop', else task will rerun
```

Задание 3. Не использовать регулярные выражения. В соответствии с заданием своего варианта составить программу для анализа текста, вводимого с клавиатуры.

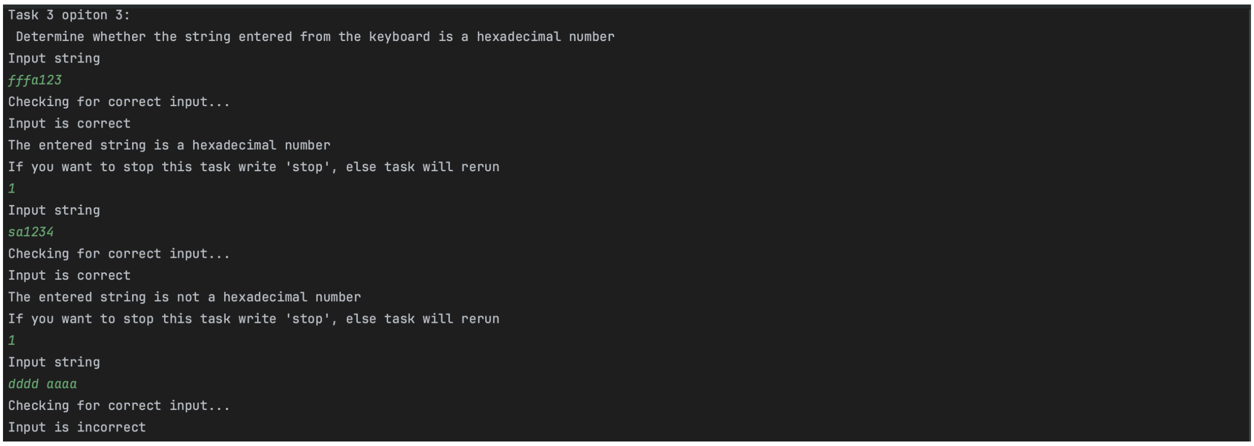
Вар-т	Условие
	Определить, является ли введенная с клавиатуры строка шестнадцатеричным числом



```python
def is_16_system(stroke):
    """doc"""
    try:
        int(stroke, 16)
        return True
    except ValueError:
        return False
def check_enter(stroke):
    """doc"""
    if stroke.find(' ')==len(stroke)-1:
        return -1
    return stroke.find(' ')

def function_task3():
    """doc"""
    RunTask = True
    while RunTask==True:
        stroke = InputStr()
        result = is_16_system(stroke)
        if result :
            print("The entered string is a hexadecimal number")
        else:
            print("The entered string is not a hexadecimal number")
        print ("If you want to stop this task write 'stop', else task will rerun")
        str = InputStop()
        if str == False:
```

Выполнение:



```
Task 3 opiton 3:
 Determine whether the string entered from the keyboard is a hexadecimal number
Input string
fffa123
Checking for correct input...
Input is correct
The entered string is a hexadecimal number
If you want to stop this task write 'stop', else task will rerun
1
Input string
sa1234
Checking for correct input...
Input is correct
The entered string is not a hexadecimal number
If you want to stop this task write 'stop', else task will rerun
1
Input string
dddd aaaa
Checking for correct input...
Input is incorrect
```

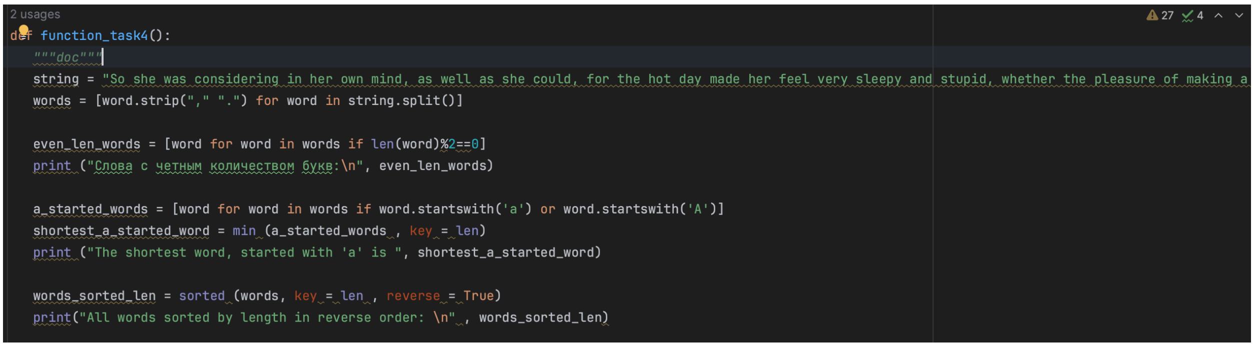
Задание 4. Не использовать регулярные выражения. Дана строка текста, в которой слова разделены пробелами и запятыми. В соответствии с заданием своего варианта составьте программу для анализа строки, инициализированной в коде программы:

«So she was considering in her own mind, as well as she could, for the hot day made her feel very sleepy and stupid, whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.»

Если не оговорено иное, то регистр букв при решении задачи не имеет значения.

Вар-т	Условие

	а) определить количество слов в строке и вывести на экран все
	слова, количество букв у которых четное;
	б) найти самое короткое слово, которое начинается на 'a';
	в) вывести повторяющиеся слова



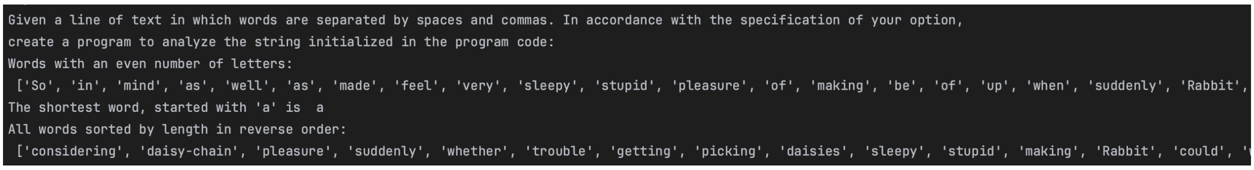
```
def function_task4():
    """doc"""
    string = "So she was considering in her own mind, as well as she could, for the hot day made her feel very sleepy and stupid, whether the pleasure of making a
    words = [word.strip("," ".") for word in string.split()]

    even_len_words = [word for word in words if len(word)%2==0]
    print ("Слова с четным количеством букв:\n", even_len_words)

    a_started_words = [word for word in words if word.startswith('a') or word.startswith('A')]
    shortest_a_started_word = min (a_started_words , key = len)
    print ("The shortest word, started with 'a' is ", shortest_a_started_word)

    words_sorted_len = sorted (words, key = len , reverse = True)
    print("All words sorted by length in reverse order: \n" , words_sorted_len)
```

Выполнение:



```
Given a line of text in which words are separated by spaces and commas. In accordance with the specification of your option,
create a program to analyze the string initialized in the program code:
Words with an even number of letters:
 ['So', 'in', 'mind', 'as', 'well', 'as', 'made', 'feel', 'very', 'sleepy', 'stupid', 'pleasure', 'of', 'making', 'be', 'of', 'up', 'when', 'suddenly', 'Rabbit',
The shortest word, started with 'a' is  a
All words sorted by length in reverse order:
 ['considering', 'daisy-chain', 'pleasure', 'suddenly', 'whether', 'trouble', 'getting', 'picking', 'daisies', 'sleepy', 'stupid', 'making', 'Rabbit', 'could', '
```

Задание 5. В соответствии с заданием своего варианта составить программу для обработки вещественных списков. Программа должна содержать следующие базовые функции:

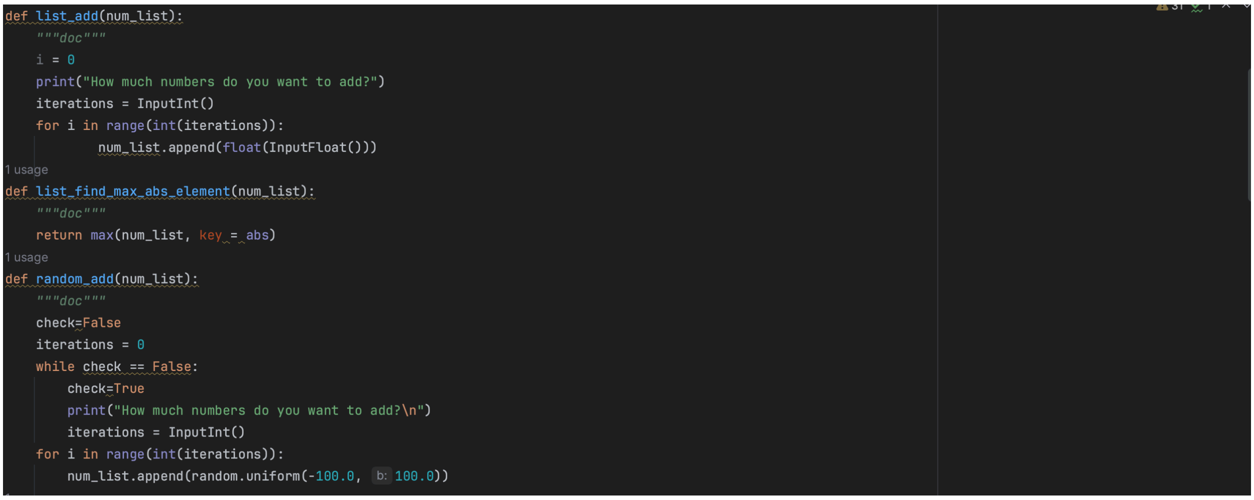
1) ввод элементов списка пользователем;

2) проверка корректности вводимых данных;

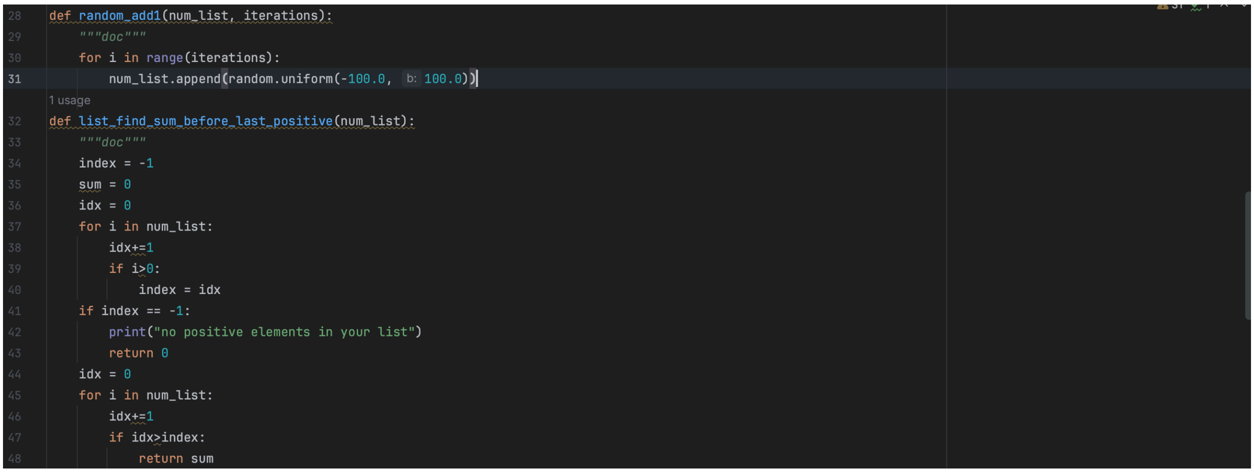
3) реализация основного задания с выводом результатов;

4) вывод списка на экран.

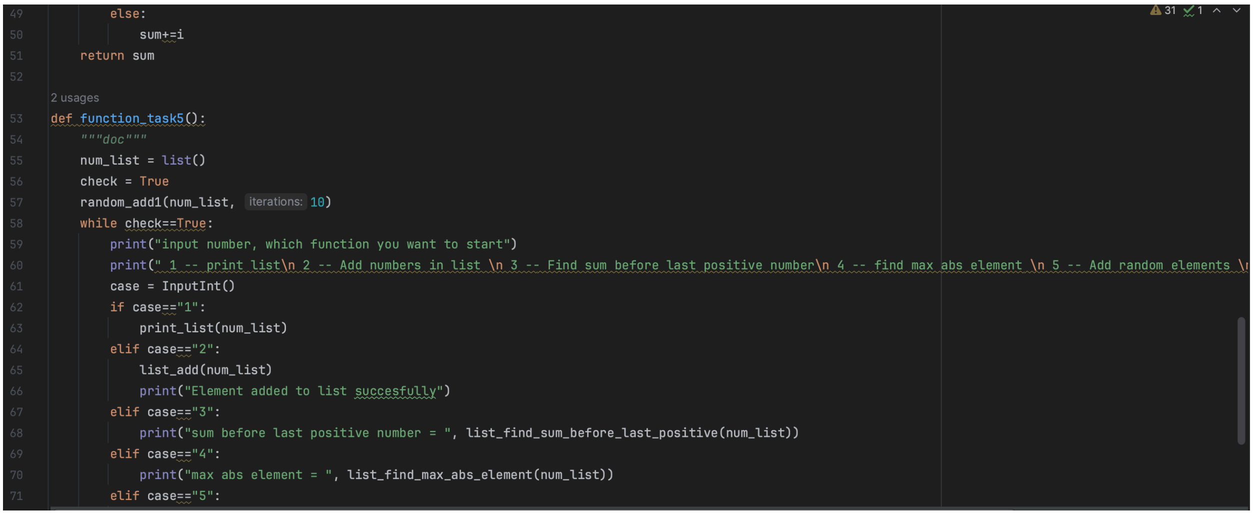
Вар	Условие
	Найти максимальный по модулю элемент списка и сумму элементов списка расположенных до последнего положительного элемента



```
def list_add(num_list):
    """doc"""
    i = 0
    print("How much numbers do you want to add?")
    iterations = InputInt()
    for i in range(int(iterations)):
            num_list.append(float(InputFloat()))
1 usage
def list_find_max_abs_element(num_list):
    """doc"""
    return max(num_list, key = abs)
1 usage
def random_add(num_list):
    """doc"""
    check=False
    iterations = 0
    while check == False:
        check=True
        print("How much numbers do you want to add?\n")
        iterations = InputInt()
    for i in range(int(iterations)):
        num_list.append(random.uniform(-100.0, b: 100.0))
```

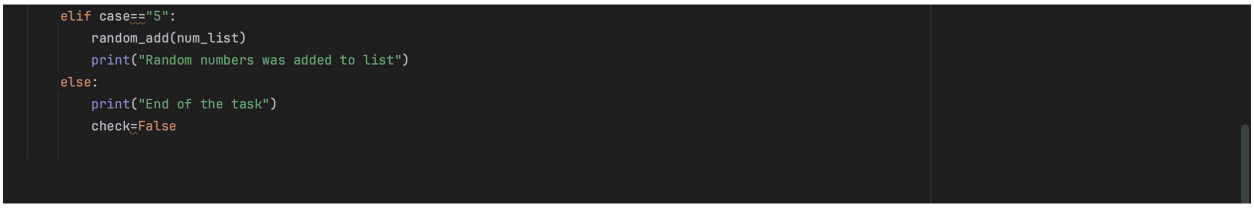


```
def random_add1(num_list, iterations):
    """doc"""
    for i in range(iterations):
        num_list.append(random.uniform(-100.0, b: 100.0))
1 usage
def list_find_sum_before_last_positive(num_list):
    """doc"""
    index = -1
    sum = 0
    idx = 0
    for i in num_list:
        idx+=1
        if i>0:
            index = idx
    if index == -1:
        print("no positive elements in your list")
        return 0
    idx = 0
    for i in num_list:
        idx+=1
        if idx>index:
            return sum
```



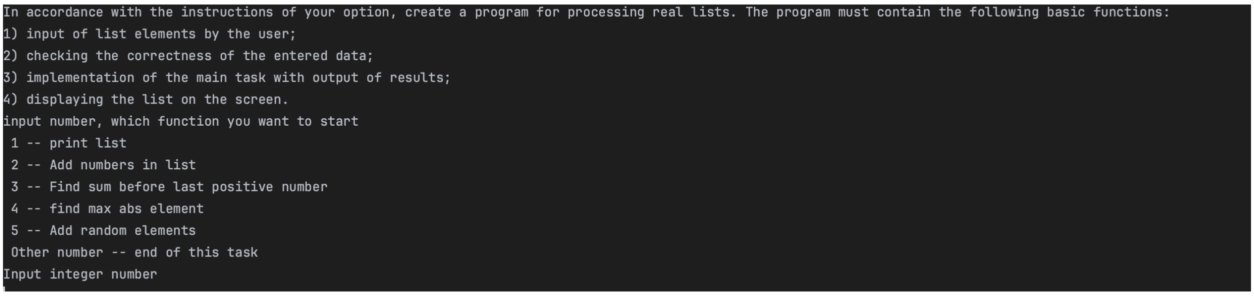
```
        else:
            sum+=i
    return sum

2 usages
def function_task5():
    """doc"""
    num_list = list()
    check = True
    random_add1(num_list, iterations: 10)
    while check==True:
        print("input number, which function you want to start")
        print(" 1 -- print list\n 2 -- Add numbers in list \n 3 -- Find sum before last positive number\n 4 -- find max abs element \n 5 -- Add random elements \n
        case = InputInt()
        if case=="1":
            print_list(num_list)
        elif case=="2":
            list_add(num_list)
            print("Element added to list succesfully")
        elif case=="3":
            print("sum before last positive number = ", list_find_sum_before_last_positive(num_list))
        elif case=="4":
            print("max abs element = ", list_find_max_abs_element(num_list))
        elif case=="5":
```

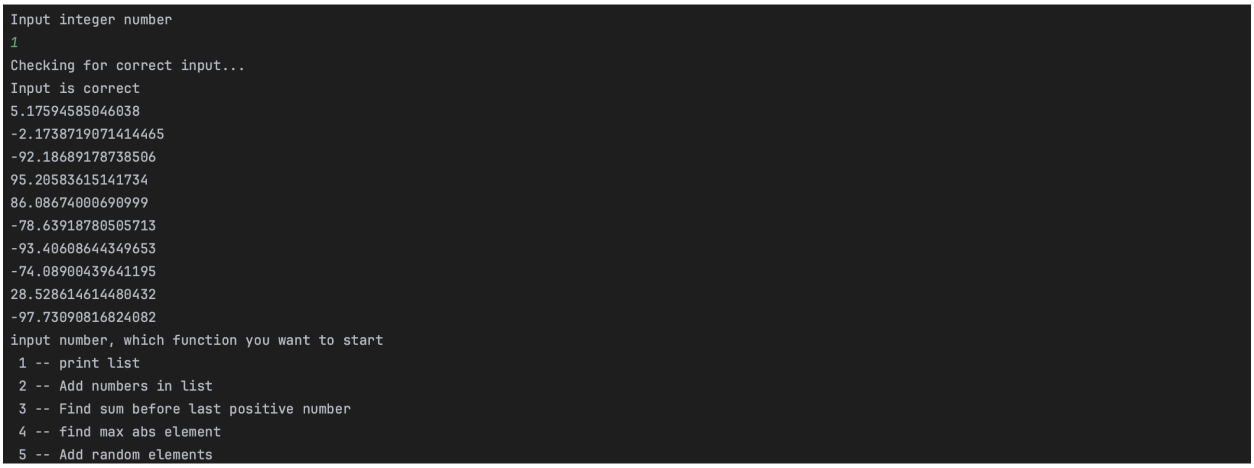


```
        elif case=="5":
            random_add(num_list)
            print("Random numbers was added to list")
        else:
            print("End of the task")
            check=False
```

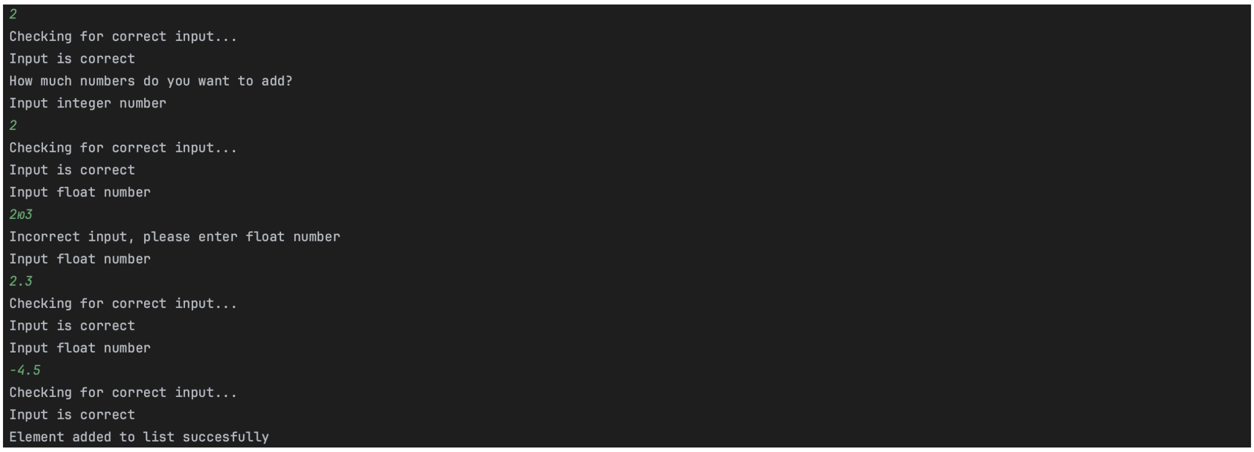
Выполнение:



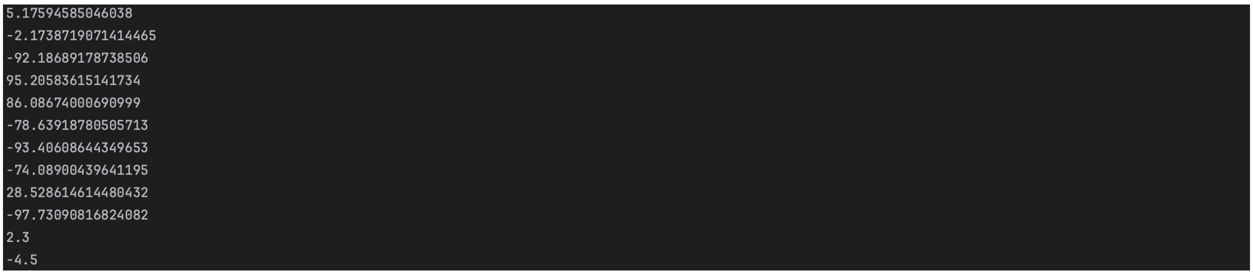
```
In accordance with the instructions of your option, create a program for processing real lists. The program must contain the following basic functions:
1) input of list elements by the user;
2) checking the correctness of the entered data;
3) implementation of the main task with output of results;
4) displaying the list on the screen.
input number, which function you want to start
 1 -- print list
 2 -- Add numbers in list
 3 -- Find sum before last positive number
 4 -- find max abs element
 5 -- Add random elements
 Other number -- end of this task
Input integer number
```



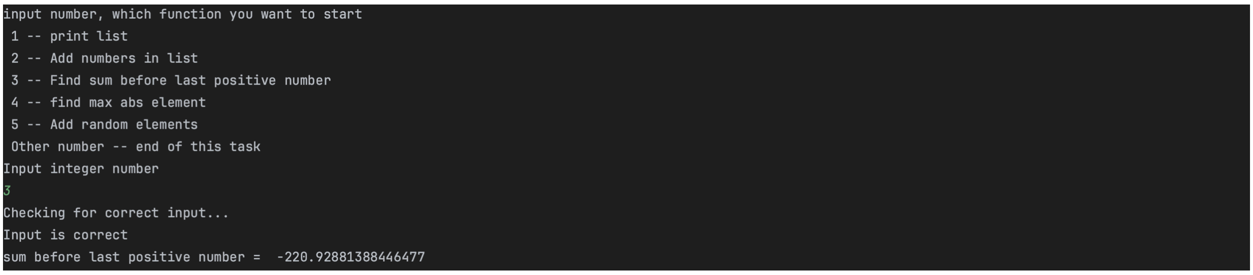
```
Input integer number
1
Checking for correct input...
Input is correct
5.17594585046038
-2.1738719071414465
-92.18689178738506
95.20583615141734
86.08674000690999
-78.63918780505713
-93.40608644349653
-74.08900439641195
28.528614614480432
-97.73090816824082
input number, which function you want to start
 1 -- print list
 2 -- Add numbers in list
 3 -- Find sum before last positive number
 4 -- find max abs element
 5 -- Add random elements
```



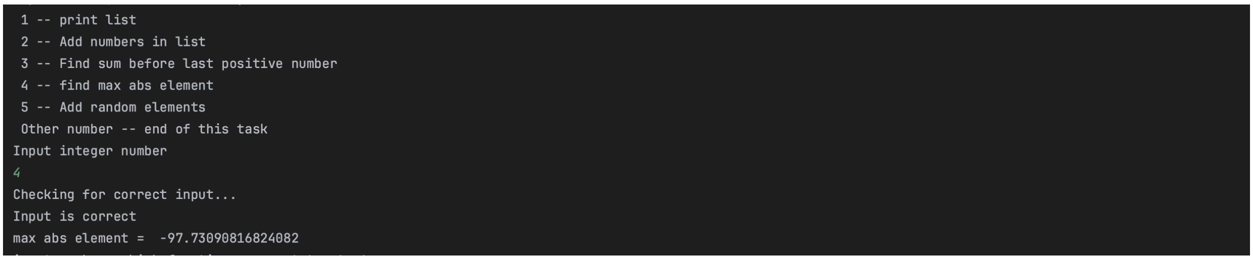
```
2
Checking for correct input...
Input is correct
How much numbers do you want to add?
Input integer number
2
Checking for correct input...
Input is correct
Input float number
2ю3
Incorrect input, please enter float number
Input float number
2.3
Checking for correct input...
Input is correct
Input float number
-4.5
Checking for correct input...
Input is correct
Element added to list succesfully
```



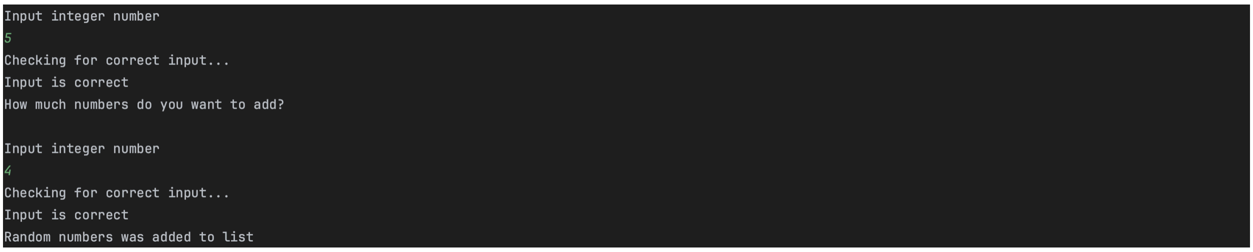
```
5.17594585046038
-2.1738719071414465
-92.18689178738506
95.20583615141734
86.08674000690999
-78.63918780505713
-93.40608644349653
-74.08900439641195
28.528614614480432
-97.73090816824082
2.3
-4.5
```



```
input number, which function you want to start
 1 -- print list
 2 -- Add numbers in list
 3 -- Find sum before last positive number
 4 -- find max abs element
 5 -- Add random elements
 Other number -- end of this task
Input integer number
3
Checking for correct input...
Input is correct
sum before last positive number =  -220.92881388446477
```



```
  1 -- print list
  2 -- Add numbers in list
  3 -- Find sum before last positive number
  4 -- find max abs element
  5 -- Add random elements
  Other number -- end of this task
 Input integer number
 4
 Checking for correct input...
 Input is correct
 max abs element =  -97.73090816824082
```



```
Input integer number
5
Checking for correct input...
Input is correct
How much numbers do you want to add?

Input integer number
4
Checking for correct input...
Input is correct
Random numbers was added to list
```