Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Избранные главы информатики»

**ОТЧЕТ**

к лабораторной работе №3

на тему:

**«Стандартные типы данных, коллекции, функции, модули Python»**

БГУИР 1-40-04-01

|  |
| --- |
| Выполнил студент группы 253504  Дмитрук Богдан Ярославович |
|  |
| (дата, подпись студента) |
| Проверила  Жвакина Анна Васильевна |
|  |
| (дата, подпись преподавателя) |

Минск 2023

**Задание 1.** В соответствии с заданием своего варианта составить программу для вычисления значения функции c 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:



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



**Программная реализация:**

import math as m

#The program allows you to calculate the value of the cos(x) function

#with a given accuracy using library functions and by decomposition into a mathematical series.

#The program outputs to the console the number of iterations required

#to achieve the specified accuracy, the passed function argument, the passed accuracy and the calculation results

def calculate\_row(x : float, eps : float):

"""

Calculats cos value through a series

Input(float x, float eps)

Output(float result, u\_int iterations number)

"""

result = 0

prev\_result = 2147483647

loop\_counter = 0

while(abs(result - prev\_result) > eps):

prev\_result = result

result += m.pow(-1, loop\_counter) \* m.pow(x, 2 \* loop\_counter) / (m.factorial(2 \* loop\_counter))

loop\_counter += 1

if loop\_counter > 500:

raise Exception("Iterations c")

return(result, loop\_counter)

def executable\_function():

"""

User interface function

Performs data input and output

"""

while(True):

try:

eps = float(input("Input eps value: "))

x = float(input("Input x value: "))

break

except ValueError:

print("\nIncorrect input!\nTry again\n")

lib\_result = m.cos(x)

my\_result, n = calculate\_row(x, eps)

print(f"x: {x}\nn: {n}\nF(x): {my\_result}\nMath F(x): {lib\_result}\neps: {eps}")

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

|  |  |
| --- | --- |
| 7. | Организовать цикл, который принимает целые числа с клавиатуры  и подсчитывает количество чисел, меньших числа 10. Окончание  цикла – ввод числа 100 |

**Программная реализация:**

#This program allows you to calculate

#the sum of terms of the integer number series and

#the number of members of series less than ten

def calculate\_function(number\_list : list[int]):

"""

Calculates amount of series numbers less 10 and sum of series

Input(list of ints)

Output(int sum\_of\_list members, u\_int less\_10\_ampount)

"""

less\_ten = len([number for number in number\_list if number < 10])

list\_sum = sum(number\_list)

return list\_sum, less\_ten

def executable\_function():

"""

User interface function

Performs data input and output

"""

number\_list = []

print("Please enter the values of the set of numbers\nTo finish entering, enter \"100\"")

while(True):

try:

number = int(input())

if(number == 100):

break

number\_list.append(number)

except ValueError:

print("\nIncorrect input!\nTry again\n")

list\_sum, less\_ten = calculate\_function(number\_list)

print(f"Sum of terms of the number series: {list\_sum}\nNumber of members of series less than ten: {less\_ten}")

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

|  |  |
| --- | --- |
| 7. | В строке, вводимой с клавиатуры, подсчитать количество букв  нижнего регистра и цифр |

**Программная реализация:**

#This program allows you to

#Calculates amount of lowcase letters and digits in passed string

def my\_decorator(func):

def wrapper():

print("main string handker function is running")

func()

return wrapper

def parse\_string(str : str):

"""

Calculates amount of lowcase letters and digits in passed string

Input(string)

Output(u\_int lowercase\_letters\_count, u\_int digits\_count)

"""

lowercase\_letters\_count = 0

digits\_count = 0

for char in str:

lowercase\_letters\_count += char.islower()

digits\_count += char.isdigit()

return lowercase\_letters\_count, digits\_count

@my\_decorator

def executable\_function():

"""

User interface function

Performs data input and output

"""

str = input("Please enter the string:\n")

lowercase\_letters\_count, digits\_count = parse\_string(str)

print(f"Lowercase letters count: {lowercase\_letters\_count}\nDigits count: {digits\_count}")

**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.

|  |  |
| --- | --- |
| 7. | а) определить число слов, которые начинаются с согласной;  б) найти слова, содержащие две одинаковые буквы подряд и их  порядковые номера;  в) вывести слова в алфавитном порядке |

**Программная реализация:**

import re

#This program allows you to

#Calculate number of words that begin with a consonant in the string

#Find words containing two identical letters in a row and their serial numbers in the string

#Sort words in string in alphabetical order in string

def count\_consonant\_start\_words(string):

"""

Calculates number of words that begin with a consonant

"""

pattern = r'\b[^aeiouAEIOU\W]\w\*\b'

matches = re.findall(pattern, string)

return len(matches)

def find\_words\_with\_duplicates(string):

"""

Finds words containing two identical letters in a row and their serial numbers

Rerturns List[Tuple(word, it's index)]

"""

answer = []

words = string.split()

for word in words:

pattern = r"(.)\1"

match = re.search(pattern, word)

if(bool(match)):

answer.append((word, words.index(word) + 1))

#pattern = r'\b((\w)\1+\b'

#matches = re.finditer(pattern, string)

#result = [(match.group(0), string[:match.start()].count(' ') + 1) for match in matches]

return answer

def sort\_words\_alphabetically(string):

"""

Sorts words in string in alphabetical order

"""

pattern = r'\b\w+\b'

matches = re.findall(pattern, string)

sorted\_words = sorted(matches)

return sorted\_words

def executable\_function():

str = "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."

print(f"\nNumber of words that begin with a consonant: \n{count\_consonant\_start\_words(str)}\n" +

f"\nWords with duplicates and their index: \n{find\_words\_with\_duplicates(str)}\n" +

f"\nSorted string: \n{sort\_words\_alphabetically(str)}")

**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.

|  |  |
| --- | --- |
|  | Найти количество положительных элементов списка, больших числа C (параметр C вводится с клавиатуры пользователем) и произведение элементов списка, расположенных после максимального по модулю элемента |

**Программная реализация:**

from functools import reduce

#This program allows you to calculate

#amount of series numbers bigger passed top border

#and product of list of elements located after the maximum modulo element

def calculate\_function(number\_list : list[float], top\_border : float):

"""

Calculates amount of series numbers bigger passer top border

and product of list of elements located after the maximum modulo element

Input(list of floats, float top\_border)

Output(u\_int bigger\_than\_count, float product)

"""

bigger\_than\_count = len([number for number in number\_list if (number > top\_border) and (number > 0)])

max\_abs\_number\_index = number\_list.index(max(number\_list, key=abs))

product = reduce(lambda x, y: x \* y, number\_list[max\_abs\_number\_index + 1 : ])

return bigger\_than\_count, product

def executable\_function():

"""

User interface function

Performs data input and output

"""

number\_list = []

print("Please enter the values of the set of numbers\nTo finish entering, enter \"100\"")

while(True):

try:

number = float(input())

if(number == 100):

break

number\_list.append(number)

except ValueError:

print("\nIncorrect input!\nTry again\n")

while(True):

try:

top\_border = float(input("Input top border:\n"))

break

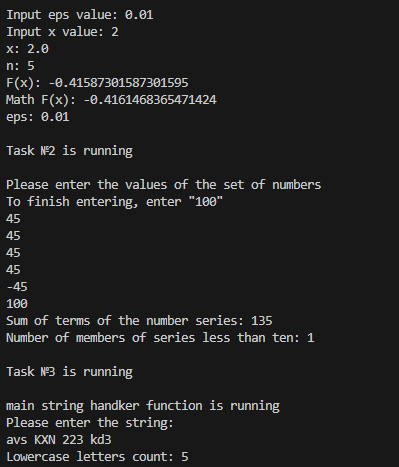
except ValueError:

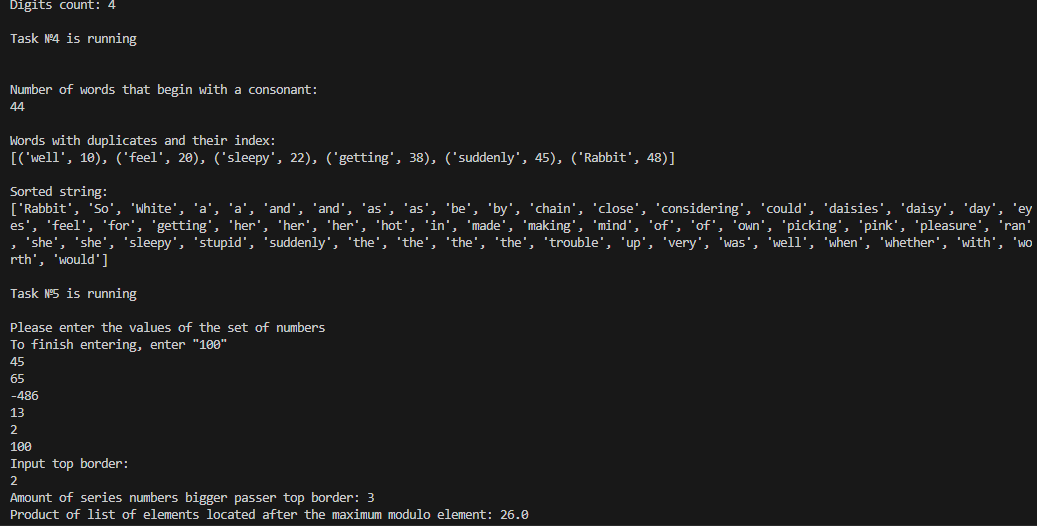
print("\nIncorrect input!\nTry again\n")

bigger\_than\_count, product = calculate\_function(number\_list, top\_border)

print(f"Amount of series numbers bigger passer top border: {bigger\_than\_count}\nProduct of list of elements located after the maximum modulo element: {product}")

**Общий лог всех реализованных модулей:**

****

****