**Лабораторная работа №5**

**Использование подпрограмм**

**Задачи:**

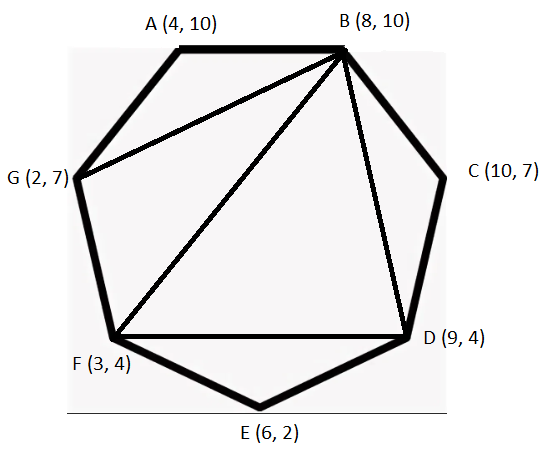
1. Изучить порядок описания подпрограмм на языке Python.

2. Научиться использовать глобальные и локальные переменные при применении подпрограмм.

3. Решить задача, составив алгоритмы и программы.

**Задача №1.**

На плоскости задан правильный семиугольник с вершинами A1(x1, y1) , A2(x2, y2) , …, A7(x7, y7). Найдите его площадь. Для решения использовать подпрограммы для вычисления длин сторон и вычисления площади треугольников



***Код программы:***

import math

def leng (x1, x2, y1, y2):

import math

l = math.sqrt(math.pow((x1 - x2), 2) + math.pow((y1 - y2), 2))

return l

def tetra (A, B, C):

import math

p = (A + B + C) / 2

s = math.sqrt(p \* (p - A) \* (p - B) \* (p - C))

return s

Ax= float(input("Enter x1: "))

Ay= float(input("Enter y1: "))

Bx= float(input("Enter x2: "))

By= float(input("Enter y2: "))

Cx= float(input("Enter x3: "))

Cy= float(input("Enter y3: "))

Dx= float(input("Enter x4: "))

Dy= float(input("Enter y4: "))

Ex= float(input("Enter x5: "))

Ey= float(input("Enter y5: "))

Fx= float(input("Enter x6: "))

Fy= float(input("Enter y6: "))

Gx= float(input("Enter x7: "))

Gy= float(input("Enter y7: "))

A = leng(Ax, Bx, Ay, By)

B = leng(Bx, Cx, By, Cy)

C = leng(Cx, Dx, Cy, Dy)

D = leng(Dx, Ex, Dy, Ey)

E = leng(Ex, Fx, Ey, Fy)

F = leng(Fx, Gx, Fy, Gy)

G = leng(Gx, Ax, Gy, Ay)

BG = leng(Bx, Gx, By, Gy)

BD = leng(Bx, Dx, By, Dy)

BF = leng(Bx, Fx, By, Fy)

FD = leng(Bx, Dx, Dy, Dx)

S = 0

S = tetra(A, G, BG) + S

S = tetra(B, C, BD) + S

S = tetra(D, E, FD) + S

S = tetra(F, BG, BF) + S

S = tetra(FD, BD, BF) + S

print(S)

*Тестирование:*

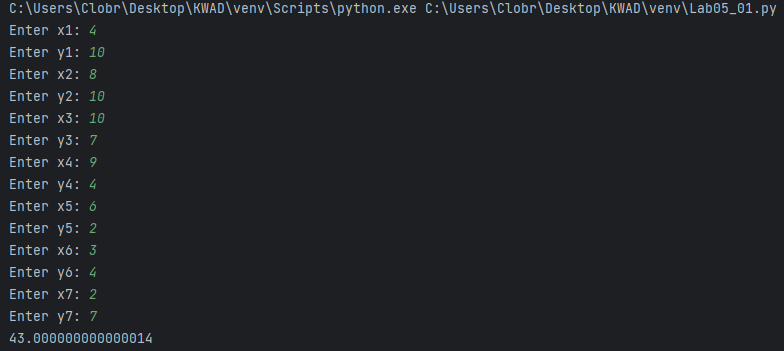


Рисунок 1.

**Задача №2.**

Вычислить , где g(x, y) = x -заданный массив.

***Код программы:***

from array import array

import math

def fun (n):

x = n + 1

y = n/2

g = math.pow(x, 2) + x \* y + math.pow(y, 2)

return g

array = array('i', [])

for i in range(15):

print ("a -",i+1)

n = int(input())

array.append(n)

g = 0

for i in range(15):

g = g + fun(array[i])

print (g)

*Тестирование:*

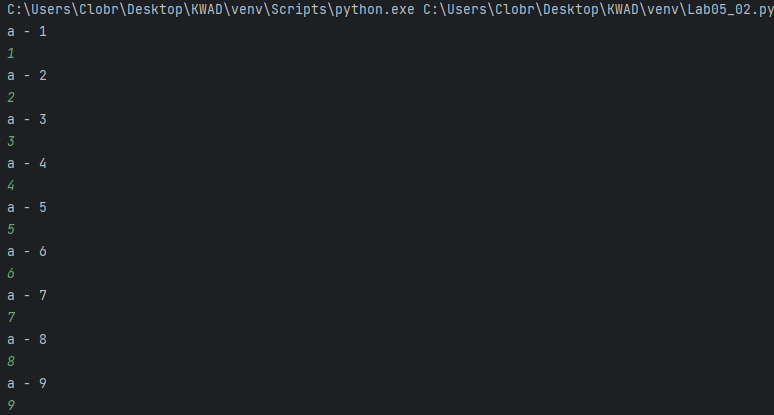




Рисунок 2.

**Задача №3.**

Напишите подпрограмму, которая анализирует поток вводимых слов, выделяет слова, которые начинаются и заканчиваются гласной, замещает в таких словах все буквы, кроме первой и последней на знак \*. Слова вводятся, пока не введено слово «стоп».

***Код программы:***

from array import array

import math

def i (array\_word, size):

array\_stop = list("стоп")

array\_glas = list("ауоыиэяюёе")

if array\_word == array\_stop:

pr = 0

return pr

else:

if array\_word[0] in array\_glas or array\_word[size - 1] in array\_glas:

for i in range(1, size - 1, 1):

array\_word[i] = "\*"

print("".join(array\_word))

else:

print("".join(array\_word))

pr = 1

return pr

pr = 1

array\_word = []

while pr == 1:

print("Введите слово: ")

word = input()

array\_word = list(word)

size = len(array\_word)

pr = i(array\_word, size)

*Тестирование:*



Рисунок 3.

**Задача №4.**

Напишите подпрограмму, которая для каждой строки матрицы А, размером m×n, находит наибольший элемент. Для подпрограммы в качестве входного параметра взять номер строки. Функцию max не использовать. Все найденные наибольшие элементы напечатать с указанием строк, к которым они относятся.

***Код программы:***

def max\_matrix(matrix, magnum, M, N):

for i in range(M):

for j in range(N):

if matr[0] <= matrix[i][j]:

matr[0] = matrix[i][j]

return matr[0]

print("Введите размер матрицы M x N")

M = int(input())

N = int(input())

print("Введите элементы матрицы")

matrix = []

for i in range(M):

row = []

for j in range(N):

el = int(input())

row.append(el)

matrix.append(row)

matr = matrix[0]

max\_matrix(matrix, matr, M, N)

print("Максимальный элемент: ", matr[0])

*Тестирование:*

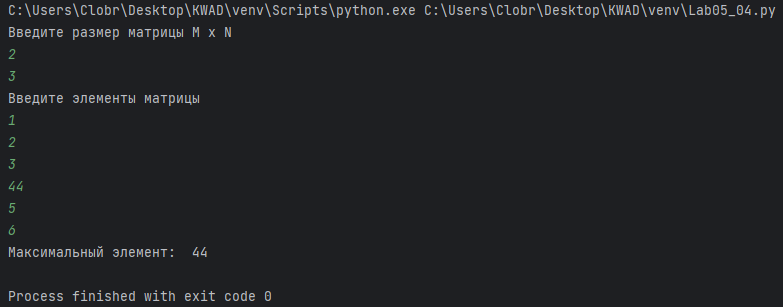


Рисунок 4.

**Задача №5.**

Подпрограмма занимается шифрованием слов, вводимых с экрана. Правило шифрования следующее. Каждый символ замещается на число - порядковый номер буквы в алфавите, к которому прибавлено значение остатка от деления порядкового номера буквы на 3. Исходные данные для подпрограммы –слово, а результат зашифрованное слово. Для работы основная программа получаем текстовую строку, слова которые разделены не более, чем одним пробелом. Пусть исходная строка состоит только из строчных букв русского алфавита. Выводит результат шифрования в виде строки. Пробелы шифровке не подлежат. Указание: использовать функцию ord (), которая возвращает код символа. Например, строка кода print (ord ('A')) напечатает число 65 – код символа «A». Аргумент функции ord () символ. Функция, которая по числу возвращает символ с данным кодом называется chr ().

***Код программы:***

def code (codes):

word2 = []

for i in codes:

a = ord(i)

a = a + (a % 3)

b = chr(a)

word2.append(b)

return word2

word = list(input("Введите строку, которую нужно зашифровать:\n"))

print (code(word))

*Тестирование:*

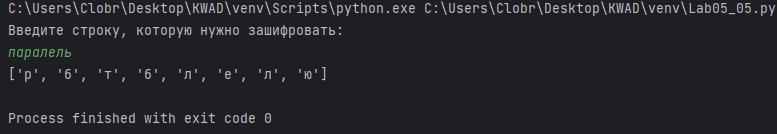


Рисунок 5.

**Задача №6.**

Пусть задано произвольное натуральное число N. Находят число - сумму его цифр. Потом для полученного числа находят сумму его цифр. Так, продолжают до тех пор, пока не получится однозначное число. Такое число принято называть цифровым корнем N. Найти для заданного N цифровой корень.

***Код программы:***

def nums (n):

l = list(n)

s = 0

for i in l:

numb = int(i)

s = s + numb

return s

n = str(input("Введите число:\n"))

s = nums(n)

while s > 9:

s = nums(str(s))

print(s)

*Тестирование:*

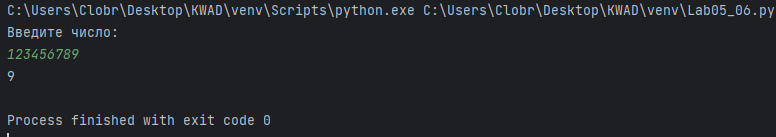


Рисунок 6.

Выполнил студент Пузанов В. Е., ФИТУ 010304-КМСб-о22

Проверил ст. преподаватель каф. ПМ Лобова Т.В.