Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «ЯП»

Тема: «Основы языка программирования Python»

Выполнил:

Студент 2 курса

Группы ПО-7(2)

Рекун И.В.

Проверил:

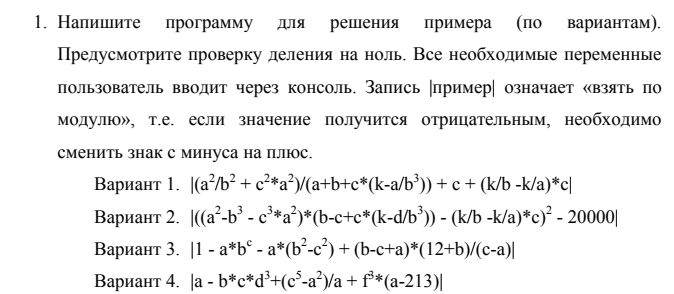
Бойко Д.О.

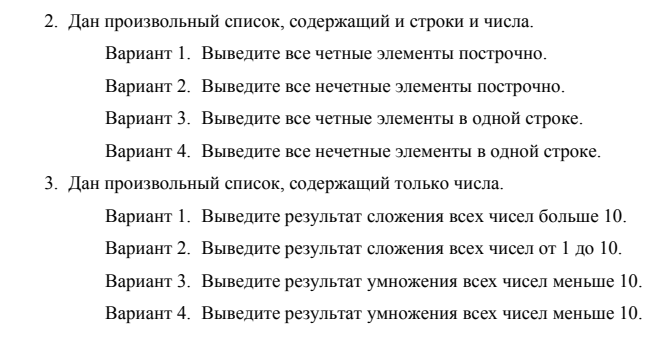
2021

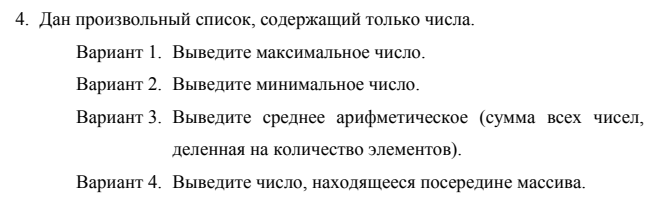
Цель работы: изучить основы языка Python.

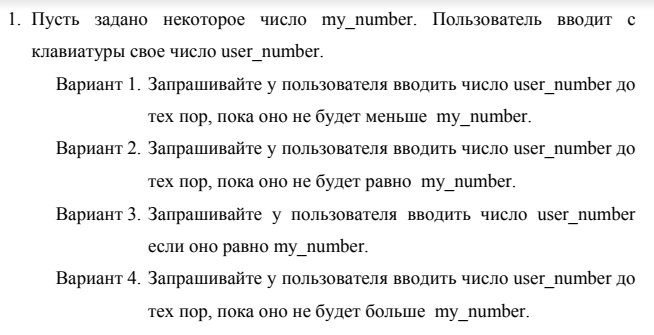
Вариант 2(по списку 6)

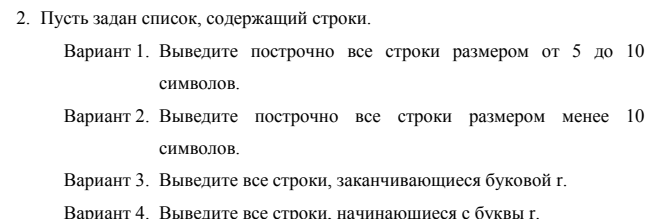
Задание:

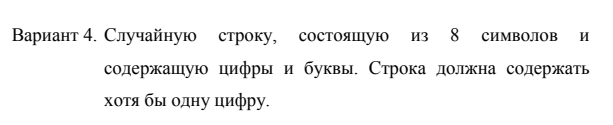


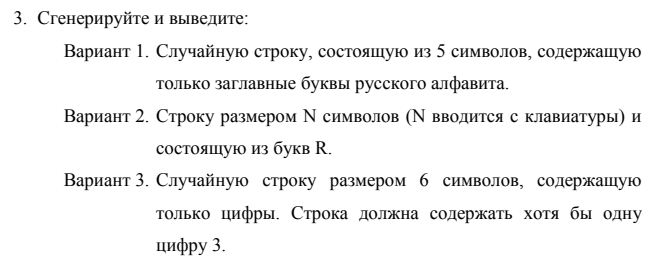


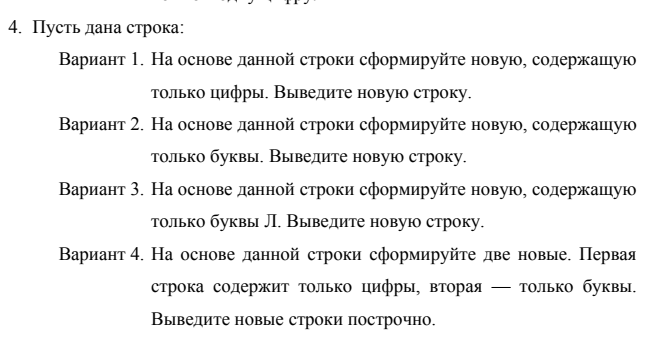


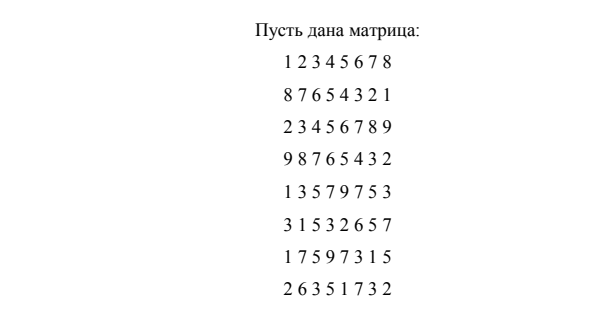


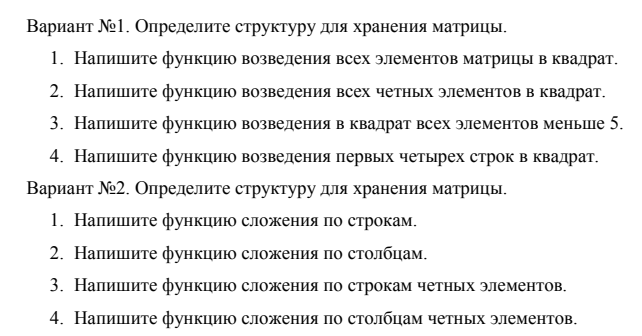


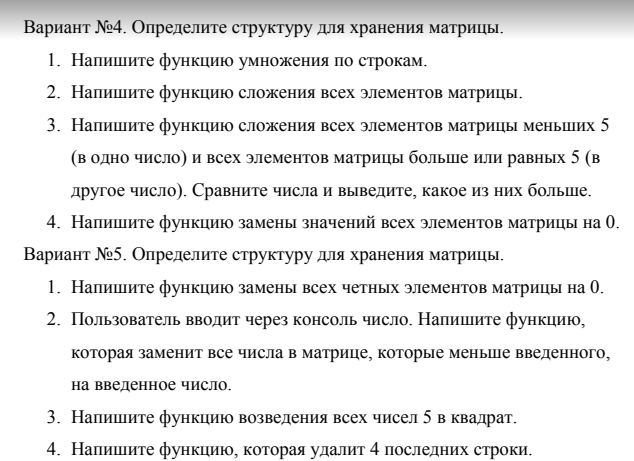


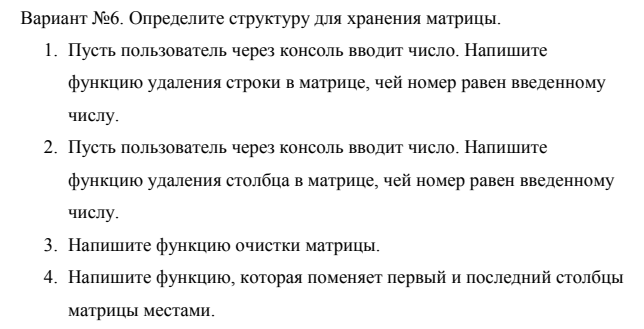


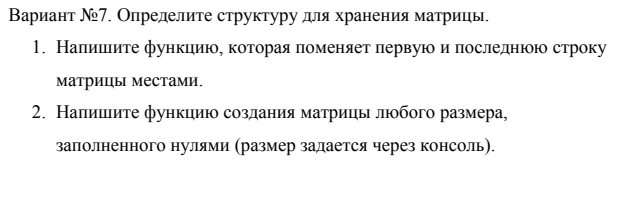


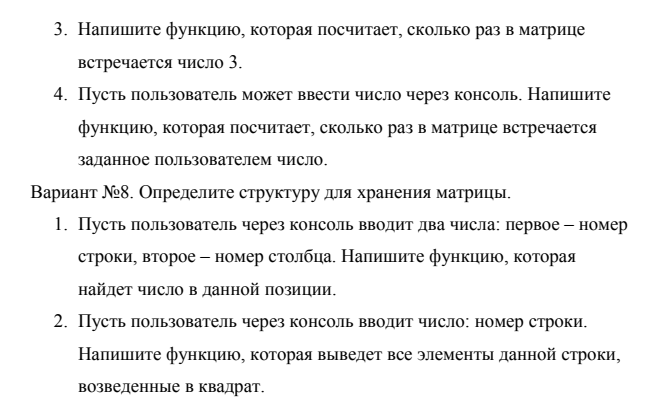


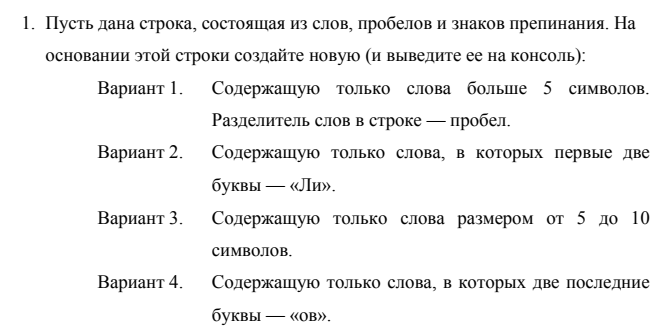


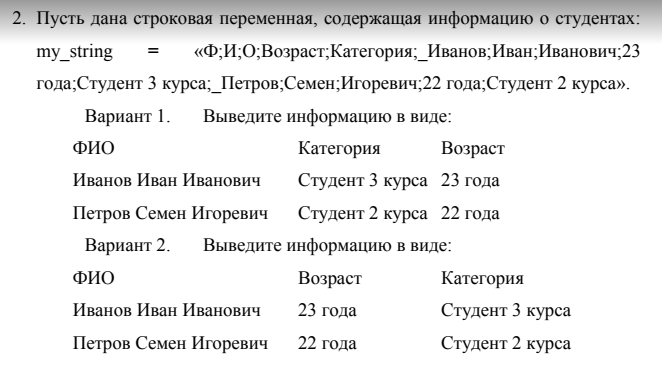


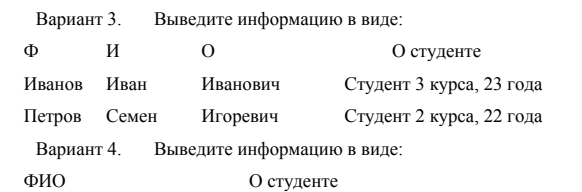


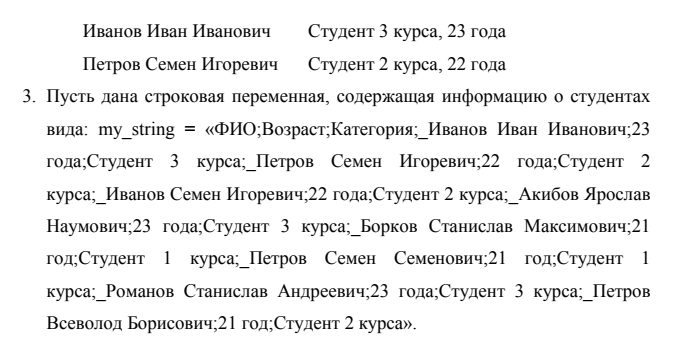


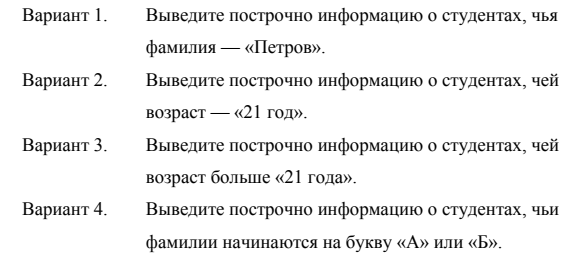


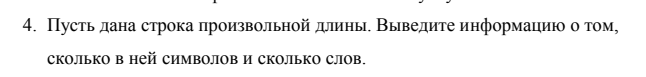


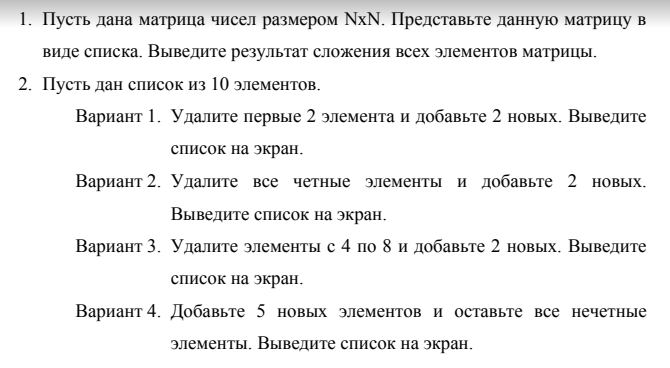


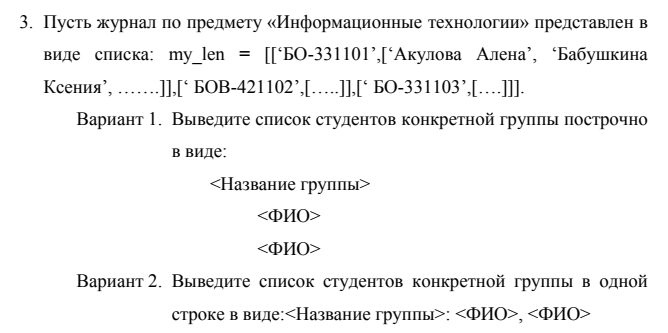


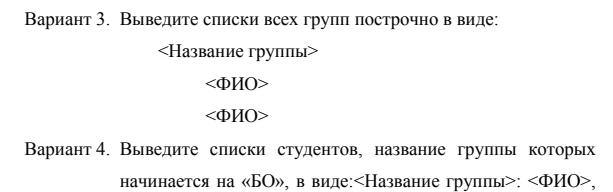


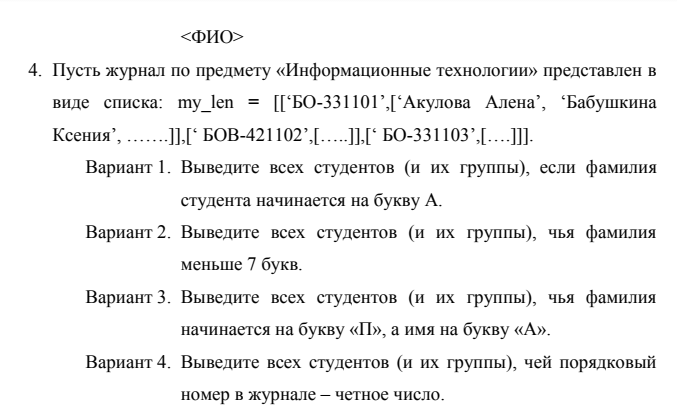












Код программы:

import random

import pandas as pd

import numpy as np

import string

import re

#z1 решение и замена отрицательного ответа положительным

def task1\_1():

a = int(input('Введите значение А: '))

if a != 0:

b = int(input('Введите значение В: '))

if b != 0:

c = int(input('Введите значение С: '))

if c != 0:

d = int(input('Введите значение D: '))

if d != 0:

k = int(input('Введите значение К: '))

if k != 0:

n = abs(((((a\*\*2) - (b\*\*3) - (c\*\*3) \* (a\*\*2))\*(b-c+c\*(k-d/(b\*\*3))))-(k/b - k/a)\*c)\*\*2 - 20000)

print(n)

#z1.2 вывести нечётные элементы

def task1\_2():

list\_a = []

for i in range(10):

list\_a.append(random.randint(0, 10))

print(list\_a)

for i in list\_a:

if i%2 == 1:

print(i)

#z1.3 вывести резултат сложения всех чисел от 1 до 10

def task1\_3():

set = []

sum\_set = []

for i in range(20):

set.append(random.randint(0,30))

print(set)

for i in set:

if i > 0 and i < 11:

sum\_set.append(i)

print(sum\_set)

print(sum(sum\_set))

#z1.4 вывести минимальное число

def task1\_4():

sp1 = []

for i in range(20):

sp1.append(random.randint(0,100))

print(sp1)

print(min(sp1))

#z2.1 ввод user\_number пока оно не будет = my\_number

def task2\_1():

my\_number = 5

user\_number = int(input('Введите число: '))

while user\_number != my\_number:

user\_number = int(input('Введите число: '))

if user\_number == my\_number:

print('Число user\_number совпало с my\_number')

#z2.2 вывести строки до 10 символов

def task2\_2():

list\_str = ['octavia', 'hello world', 'thank you for your attention', 'McQueen']

num = 10

i = 0

while len(list\_str) > i:

if len(list\_str[i]) < num:

print(list\_str[i])

i += 1

#z2.3 строку из R размером N

def task2\_3():

n = int(input('Введите длину строки: '))

string\_R = 'R'\*n

print(string\_R)

#z2.4 дана строка, вывести новую из букв

def task2\_4():

text = [random.choice('abc123') for \_ in range(10)]

text[5] = random.choice('ABC')

random\_string = ''.join(text)

print(random\_string)

str2=''

for c in random\_string:

if c not in ('0','1','2','3','4','5','6','7','8','9'):

str2=str2+c

print(str2)

#z3.1 возвести чётные элементы матрицы в квадрат

def task3\_1():

matrix = np.array([[1, 2, 3, 4, 5, 6, 7, 8], [8, 7, 6, 5, 4, 3, 2, 1,], [2, 3, 4, 5, 6, 7, 8, 9], [9, 8, 7, 6, 5, 4, 3, 2], [1, 3, 5, 7, 9, 7, 5, 3], [3, 1, 5, 3, 2, 6, 5, 7], [1, 7, 5, 9, 7, 3, 1, 5], [2, 6, 3, 5, 1, 7, 3, 2]])

print(matrix)

set = []

for row in matrix:

for el in row:

if el%2 == 0:

set.append(el)

print(set)

set = np.array(set)

set2 = set \* set

print(set2)

#z3.2 сложение по столбцам

def task3\_2():

matrix = np.array([[1, 2, 3, 4, 5, 6, 7, 8], [8, 7, 6, 5, 4, 3, 2, 1,], [2, 3, 4, 5, 6, 7, 8, 9], [9, 8, 7, 6, 5, 4, 3, 2], [1, 3, 5, 7, 9, 7, 5, 3], [3, 1, 5, 3, 2, 6, 5, 7], [1, 7, 5, 9, 7, 3, 1, 5], [2, 6, 3, 5, 1, 7, 3, 2]])

print(matrix)

print(list(map(sum, zip(\*matrix))))

#z3.4 сложение всех элементов

def task3\_4():

matrix = np.array([[1, 2, 3, 4, 5, 6, 7, 8], [8, 7, 6, 5, 4, 3, 2, 1,], [2, 3, 4, 5, 6, 7, 8, 9], [9, 8, 7, 6, 5, 4, 3, 2], [1, 3, 5, 7, 9, 7, 5, 3], [3, 1, 5, 3, 2, 6, 5, 7], [1, 7, 5, 9, 7, 3, 1, 5], [2, 6, 3, 5, 1, 7, 3, 2]])

print(matrix)

print(np.sum(matrix))

#z3.5 заменить элементы матрицы меньше c, на c

def task3\_5():

matrix = np.array([[1, 2, 3, 4, 5, 6, 7, 8], [8, 7, 6, 5, 4, 3, 2, 1,], [2, 3, 4, 5, 6, 7, 8, 9], [9, 8, 7, 6, 5, 4, 3, 2], [1, 3, 5, 7, 9, 7, 5, 3], [3, 1, 5, 3, 2, 6, 5, 7], [1, 7, 5, 9, 7, 3, 1, 5], [2, 6, 3, 5, 1, 7, 3, 2]])

print(matrix)

c = int(input('Введите число: '))

matrix[matrix < c] = c

print(matrix)

#z3.6 удаление столбца по номеру

def task3\_6():

new\_matrix = np.random.randint(10, size=(3, 4))

print(new\_matrix)

b = int(input('Введите число: '))

new\_matrix = np.delete(new\_matrix,[b],1)

print(new\_matrix)

#z3.7 создание матрицы из 0, задаваемого размера

def task3\_7():

a = int(input('Введите количество строк: '))

b = int(input('Введите количество столбцов: '))

new\_matrix = np.zeros((a, b))

print(new\_matrix)

#z3.8 вывести строку в квадрате по номеру

def task3\_8():

new\_matrix = np.random.randint(10, size=(3, 4))

print(new\_matrix)

num\_row = int(input('Введите номер строки: '))

row1 = new\_matrix[num\_row]

row = row1\*row1

print(row)

#z4.1 вывести из строки слова на "Ли"

def task4\_1():

str = 'Лиссабон, Берлин, Ливерпуль, Гданьск, Лиман, Вроцлав, Москва, Брно, Лилль'

print(\*(word for word in str.split() if word.startswith('Ли')))

#z4.2 вывести информацию

def task4\_2():

my\_string = 'ФИО; Возраст; Категория; Иванов; Иван; Иванович; 23 года; Студент 3 курса; Петров; Семён; Игоревич; 22 года; Студент 2 курса'

spisok = pd.DataFrame({

"ФИО": ['Иванов Иван Иванович', 'Петров Семён Игоревич'],

"Возраст": ['23 года', '22 года'],

"Категория": ['Студент 3 курса', 'Студент 2 курса']

})

print(spisok)

#z4.3 вывести информацию о студентах 21 года

def task4\_3():

my\_len = (

"ФИО;Возраст;Категория;"

"\_Иванов Иван Иванович;23 года;Студент 3 курса;"

"\_Петров Семен Игоревич;22 года;Студент 2 курса;"

"\_Иванов Семен Игоревич;22 года;Студент 2 курса;"

"\_Акибов Ярослав Навич;23 года;Студент 3 курса;"

"\_Борков Станислав Максимович;21 год;Студент 1 курса;"

"\_Петров Семен Семенович;21 год;Студент 1 курса;"

)

students = my\_len.split(";\_")

data\_new = []

info = []

for row in students:

data\_new.append(row.split(";"))

for i in data\_new[1:]:

if "21" in i[1]:

info.append(i)

for i in info:

print(" ".join(i))

#z4.4 посчитать количество слов и символов в строке

def task4\_4():

s = 'Нельзя понять всю красоту леса, оценивая лишь одно дерево. Мы живём настоящим, скорбим о прошлом и гадаем о будущем. События всегда происходят без предупреждения и лишь потом становится ясна причина. Назвав месть правосудием, они лишь принесёт больше мести взамен'

print(len(s.split()))

print(len(s))

#z6.1 сумма всех элементов матрицы

def task6\_1():

matrix1 = np.random.randint(10, size=(4, 4))

print(matrix1)

print(np.sum(matrix1))

#z6.2 удалить все чётные элементы списка, и добавить 2 новых

def task6\_2():

my\_set = []

my\_set1 = []

for i in range(10):

my\_set.append(random.randint(0,100))

print(my\_set)

for i in my\_set:

if i%2 != 0:

my\_set1.append(i)

print(my\_set1)

my\_set1.append(5050)

my\_set1.append(5555)

print(my\_set1)

#z6.3 вывести инфу по группе

def task6\_3():

group = [["БО-331101", ["Акулова Алена", "Пабушкина Асения"]], ["БО-402000", ["Солышко Константин", "Солышко Дмитрий"]]]

number = 'БО-331101'

for i in group:

if number in i[0]:

print(f"{i[0]}:{', '.join(i[1])}")

#z6.4 вывести инфу о тех чья фамилия < 7

def task6\_4():

new\_group = [["БО-331101", ["Акулова Алена", "Пабушкина Асения"]],

["БО-402000", ["Шнитко Владислав", "Куц Владислав"]]]

temp = []

for i in new\_group:

for j in i[1]:

temp.append(j.split(' '))

for i in temp:

if len(i[0]) < 7:

print(' '.join(i))

#z7.2 меню для задания 1

def task7\_2():

print('Меню:')

menu = pd.Series([ '- Выход', ' - Запустить функцию №1', ' - Запустить функцию №2', ' - Запустить функцию №3', ' - Запустить функцию №4'])

print(menu)

def choice():

var = int(input('Введите номер: ' ))

if var == 1:

task1\_1()

if var == 2:

task1\_2()

if var == 3:

task1\_3()

if var == 4:

task1\_4()

if var == 0:

print('Спасибо за внимание')

choice()

while True:

why = input('Желаете продолжить(Да/Нет): ')

if why == 'Да':

choice()

if why == 'Нет':

break

def main() -> None:

print('\nЗадание №1')

print('Задание 1.1')

task1\_1()

print('Задание 1.2')

task1\_2()

print('Задание 1.3')

task1\_3()

print('Задание 1.4')

task1\_4()

print('\nЗадание №2 «Строки и списки»')

print('Задание 2.1')

task2\_1()

print('Задание 2.2')

task2\_2()

print('Задание 2.3')

task2\_3()

print('Задание 2.4')

task2\_4()

print('\nЗадание №3 «Матрицы»')

print('Задание 3.1')

task3\_1()

print('Задание 3.2')

task3\_2()

print('Задание 3.4')

task3\_4()

print('Задание 3.5')

task3\_5()

print('Задание 3.6')

task3\_6()

print('Задание 3.7')

task3\_7()

print('Задание 3.8')

task3\_8()

print('\nЗадание №4 «Строки»')

print('Задание 4.1')

task4\_1()

print('Задание 4.2')

task4\_2()

print('Задание 4.3')

task4\_3()

print('Задание 4.4')

task4\_4()

print('\nЗадание №6 «Списки»')

print('Задание 6.1')

task6\_1()

print('Задание 6.2')

task6\_2()

print('Задание 6.3')

task6\_3()

print('Задание 6.4')

task6\_4()

print('\nЗадание №7 «Функции»')

print('Задание 7.2')

task7\_2()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Вывод: В ходе выполнения данной работы я изучил базовые операции языка, научился работать со строками, списками, а также функциями.