

Практическая работа №3

по теме: «Циклический оператор»

Вариант - 9

Студента группы ИСиП-31

Калмыкова Александра

Преподаватель

Кондурар Марина Викторовна

Тольятти-2021

Оглавление

[Задание№1 4](#_Toc66392104)

[Задание№2 6](#_Toc66392105)

[Задание№3 9](#_Toc66392106)

[Задание№4 12](#_Toc66392107)

[Задание№5 15](#_Toc66392108)

[Задание№6 18](#_Toc66392109)

[Задание№7 20](#_Toc66392110)

[Задание№8 23](#_Toc66392111)

[Задание№9 26](#_Toc66392112)

[Задание№10 29](#_Toc66392113)

[Задание№11 32](#_Toc66392114)

[Задание№12 35](#_Toc66392115)

[Задание№13 38](#_Toc66392116)

[Задание№14 41](#_Toc66392117)

[Задание№15 44](#_Toc66392118)

[Задание№16 47](#_Toc66392119)

[Задание№17 50](#_Toc66392120)

[Задание№18 53](#_Toc66392121)

[Задание№19 55](#_Toc66392122)

[Задание№20 57](#_Toc66392123)

[Задание№21 59](#_Toc66392124)

[Задание№22 61](#_Toc66392125)

[Задание№23 63](#_Toc66392126)

[Задание№24 66](#_Toc66392127)

[Задание№25 69](#_Toc66392128)

[Задание№26 71](#_Toc66392129)

# Задание№1

По данному целому числу N распечатайте все квадраты натуральных чисел, не превосходящие N, в порядке возрастания.

**Входные данные:** N - int

**Математические операции:**

count = 0

count += 1

kvadrat = count\*\*2

**Выходные данные:** count, kvadrat - int

**Блок-схема:**



**Код:**

print("Введите число N...")

N = int(input())

count = 0

while count < N:

    count += 1

    print("N:",count)

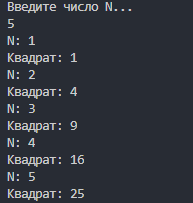
    kvadrat = count\*\*2

    print("Квадрат:",kvadrat)

**Контрольный пример:**

N = 5

**Результат выполнения программы:**



# Задание№2

Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

**Входные данные:** number - int

**Математические операции:**

i = 2

number % i != 0

i += 1

**Выходные данные:** i - int

**Блок-схема:**



**Код:**

print("Введите число не меньшее 2...")

number = int(input())

i = 2

while number % i != 0:

    i += 1

print("Наименьший делитель:",i)

**Контрольный пример:**

number = 125

**Результат выполнения программы:**



# Задание№3

По данному натуральному числу N найдите наибольшую целую степень двойки, не превосходящую N. Выведите показатель степени и саму степень. Операцией возведения в степень пользоваться нельзя!

**Входные данные:** number - int

**Математические операции:**

twoDegree = 2

degree = 1

twoDegree \*= 2

degree += 1

**Выходные данные:** twoDegree, degree - int

**Блок-схема:**



**Код:**

print("Введите число... ")

number = int(input())

twoDegree = 2

degree = 1

while twoDegree <= number:

    twoDegree \*= 2

    degree += 1

print("Полученная степень:",degree - 1, "| 2 ^",degree - 1 , "=",twoDegree // 2)

**Контрольный пример:**

number = 10

**Результат выполнения программы:**



# Задание№4

В первый день спортсмен пробежал x километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу y определите номер дня, на который пробег спортсмена составит не менее y километров. Программа получает на вход действительные числа x и y и должна вывести одно натуральное число.

**Входные данные:** x, y - int

**Математические операции:**

day = 1

x \*= 1.1

day += 1

**Выходные данные:** day - int

**Блок-схема:**

****

**Код:**

print("Введите x...")

x = int(input())

print("Введите y...")

y = int(input())

day = 1

while x < y:

    x \*= 1.1

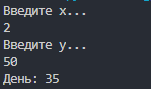
    day += 1

print("День:", day)

**Контрольный пример:**

x = 2; y = 50;

**Результат выполнения программы:**



# Задание№5

Программа получает на вход последовательность целых неотрицательных чисел, каждое число записано в отдельной строке. Последовательность завершается числом 0, при считывании которого программа должна закончить свою работу и вывести количество членов последовательности (не считая завершающего числа 0). Числа, следующие за числом 0, считывать не нужно.

**Входные данные:** number - int

**Математические операции:**

length = 0

length += 1

**Выходные данные:** length - int

**Блок-схема:**



**Код:**

length = 0

while int(input("Введите число: ")) != 0:

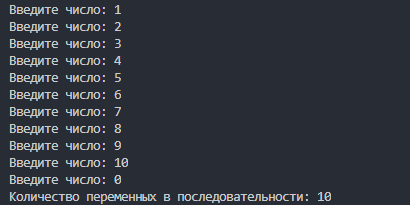
    length += 1

print("Количество переменных в последовательности:",length)

**Контрольный пример:**

number = 1, 2 , 3, … 10, 0;

**Результат выполнения программы:**

****

# Задание№6

Определите сумму всех элементов последовательности.

**Входные данные:** number - int

**Математические операции:**

counter = 0

number != 0

counter += number

**Выходные данные:** counter - int

**Блок-схема:**

****

**Код:**

counter = 0

number = int(input("Введите число: "))

while number != 0:

    counter += number

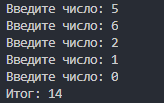
    number = int(input("Введите число: "))

print("Итог:", counter)

**Контрольный пример:**

number = 5, 6, 7, 2 ,1, 0;

**Результат выполнения программы:**

****

# Задание№7

Определите среднее значение всех элементов последовательности.

**Входные данные:** number - int

**Математические операции:**

counter = 0

count = 0

counter += number

count += 1

average = counter // count

**Выходные данные:** average – int

**Блок-схема:**



**Код:**

counter = 0

count = 0

number = int(input("Введите число: "))

while number != 0:

    counter += number

    count += 1

    number = int(input("Введите число: "))

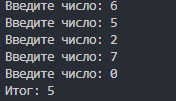
average = counter // count

print("Итог:", average)

**Контрольный пример:**

number = 6, 5, 2, 7, 0

**Результат выполнения программы:**

****

# Задание№8

Последовательность состоит из натуральных чисел и завершается числом 0. Определите значение наибольшего элемента последовательности.

**Входные данные:** number - int

**Математические операции:**

maxNum = 0

maxNum = number

**Выходные данные:** maxNum - int

**Блок-схема:**

****

**Код:**

maxNum = 0

number = int(input("Введите число: "))

while number != 0:

    number = int(input("Введите число: "))

    if(number > maxNum):

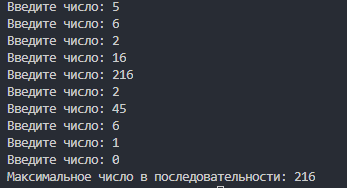
        maxNum = number

print("Максимальное число в последовательности:", maxNum)

**Контрольный пример:**

Number = 5, 6, 2, 16, 216, 2, 45, 6, 1, 0

**Результат выполнения программы:**



# Задание№9

Определите индекс наибольшего элемента последовательности. Если наибольших элементов несколько, выведите индекс первого из них. Нумерация элементов начинается с нуля.

**Входные данные:** number - int

**Математические операции:**

maxNum = 0

maxIndex = -1

number = -1

length = 0

maxNum = number

maxIndex = length

length += 1

**Выходные данные:** maxIndex, maxNum - int

**Блок-схема:**



**Код:**

maxNum = 0

maxIndex = -1

number = -1

length = 0

while number != 0:

    number = int(input("Введите число:"))

    if number > maxNum:

        maxNum = number

        maxIndex = length

    length += 1

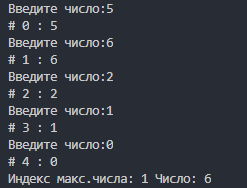
    print("#",length-1,":",number)

print("Индекс макс.числа:",maxIndex, "Число:", maxNum)

**Контрольный пример:**

number = 5, 6, 2, 1, 0

**Результат выполнения программы:**



# Задание№10

Определите количество четных элементов в последовательности.

**Входные данные:** number - int

**Математические операции:**

chet = -1

number = -1

chet += 1

**Выходные данные:** chet - int

**Блок-схема:**



**Код:**

chet = -1

number = -1

while number != 0:

    number = int(input("Введите число: "))

    if number % 2 == 0:

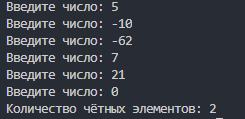
        chet += 1

print("Количество чётных элементов:",chet)

**Контрольный пример:**

number = 5, -10, -62, 7, 21, 0

**Результат выполнения программы:**

****

# Задание№11

Определите, сколько элементов этой последовательности больше предыдущего элемента.

**Входные данные:** previos, next - int

**Математические операции:**

result = 0

previos = next

**Выходные данные:** result - int

**Блок-схема:**



**Код:**

previos = int(input("Введите число: "))

result = 0

while previos != 0:

    next = int(input("Введите число: "))

    if next != 0 and previos < next:

        result += 1

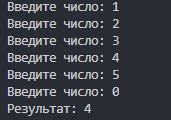
    previos = next

print("Результат:",result)

**Контрольный пример:**

Previos-next = 1, 2, 3, 4, 5, 0;

**Результат выполнения программы:**

****

# Задание№12

Последовательность состоит из различных натуральных чисел и завершается числом 0. Определите значение второго по величине элемента в этой последовательности. Гарантируется, что в последовательности есть хотя бы два элемента.

**Входные данные:** maxFirst, maxSecond, number - int

**Математические операции:**

maxFirst, maxSecond = maxSecond, maxFirst

maxSecond, maxFirst = maxFirst, number

maxSecond = number

**Выходные данные:** maxSecond - int

**Блок-схема:**

****

**Код:**

maxFirst = int(input("Введите число: "))

maxSecond = int(input("Введите число: "))

if maxFirst < maxSecond:

    maxFirst, maxSecond = maxSecond, maxFirst

number = int(input("Введите число: "))

while number != 0:

    if number > maxFirst:

        maxSecond, maxFirst = maxFirst, number

    elif number > maxSecond:

        maxSecond = number

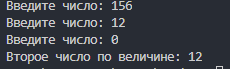
    number = int(input("Введите число: "))

print("Второе число по величине:",maxSecond)

**Контрольный пример:**

maxFirst, maxSecond, number = 156, 12, 0;

**Результат выполнения программы:**



# Задание№13

Последовательность состоит из натуральных чисел и завершается числом 0. Определите, сколько элементов этой последовательности равны ее наибольшему элементу

**Входные данные:** number – int

**Математические операции:**

maximum = 0

maximalNum = 0

number = -1

maximum, maximalNum = number, 1

maximalNum = maximalNum + 1

**Выходные данные:** maximalNum - int

**Блок-схема:**



**Код:**

maximum = 0

maximalNum = 0

number = -1

while number != 0:

    number = int(input("Введите число: "))

    if number > maximum:

        maximum, maximalNum = number, 1

    elif number == maximum:

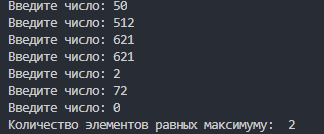
        maximalNum += 1

print("Количество элементов равных максимуму: ",maximalNum)

**Контрольный пример:**

Number = 50, 512, 621, 621, 2, 72, 0;

**Результат выполнения программы:**

****

# Задание№14

Последовательность Фибоначчи определяется так: φ0 = 0, φ1 = 1, φn = φn−1 + φn−2. По данному числу n определите n-е число Фибоначчи φn.

**Входные данные:** number - int

**Математические операции:**

a, b = 0, 1

a, b = b, a + b

**Выходные данные:** b - int

**Блок-схема:**

****

**Код:**

number = int(input("Введите число: "))

if number == 0:

    print(0)

else:

    a, b = 0, 1

    for i in range(2, number + 1):

        a, b = b, a + b

    print("Ответ:",b)

**Контрольный пример:**

number = 9;

**Результат выполнения программы:**

****

# Задание№15

Дано натуральное число *A*. Определите, каким по счету числом Фибоначчи оно является, то есть выведите такое число *n*, что *φn = A*. Если *А* не является числом Фибоначчи, выведите число -1.

**Входные данные:** number - int

**Математические операции:**

fibPrev, fibNext = 0, 1

n = 1

fibPrev, fibNext = fibNext, fibPrev + fibNext

n += 1

**Выходные данные:** n - int

**Блок-схема:**

****

**Код:**

number = int(input("Введите число: "))

if number == 0:

    print(0)

else:

    fibPrev, fibNext = 0, 1

    n = 1

    while fibNext <= number:

        if fibNext == number:

            print(n)

            break

        fibPrev, fibNext = fibNext, fibPrev + fibNext

        n += 1

    else:

        print(-1)

**Контрольный пример:**

number = 34

**Результат выполнения программы:**

****

# Задание№16

Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу.

**Входные данные:** number - int

**Математические операции:**

previos = -1

currentRepLength = 0

maxRepLength = 0

currentRepLength += 1

previos = number

maxRepLength = max(maxRepLength, currentRepLength)

currentRepLength = 1

maxRepLength = max(maxRepLength, currentRepLength)

**Выходные данные:** maxRepLength - int

**Блок-схема:**



**Код:**

previos = -1

currentRepLength = 0

maxRepLength = 0

number = int(input("Введите число: "))

while number != 0:

    if previos == number:

        currentRepLength += 1

    else:

        previos = number

        maxRepLength = max(maxRepLength, currentRepLength)

        currentRepLength = 1

    number = int(input("Введите число: "))

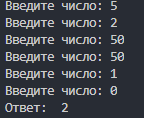
maxRepLength = max(maxRepLength, currentRepLength)

print("Ответ: ", maxRepLength)

**Контрольный пример:**

number = 2, 5, 50, 50, 1, 0;

**Результат выполнения программы:**

****

# Задание№17

Дана последовательность натуральных чисел x1, x2, ..., xn. Стандартным отклонением называется величина σ=√((x1−s) 2+(x2−s) 2+…+(xn−s) 2 )/(n−1), где s=(x1+x2+…+xn)/n — среднее арифметическое последовательности. Определите стандартное отклонение для данной последовательности натуральных чисел, завершающейся числом 0.

**Входные данные:** number – int

**Математические операции:**

partialSum = 0

partialSumSquares = 0

n = 0

n += 1

partialSum += number

partialSumSquares += number \*\* 2

answer = sqrt((partialSumSquares - partialSum \*\* 2 / n) / (n - 1))

**Выходные данные:** answer - float

**Блок-схема:**

****

**Код:**

from math import sqrt

partialSum = 0

partialSumSquares = 0

n = 0

number = int(input("Введите число: "))

while number != 0:

    n += 1

    partialSum += number

    partialSumSquares += number \*\* 2

    number = int(input("Введите число: "))

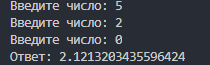
answer = sqrt((partialSumSquares - partialSum \*\* 2 / n) / (n - 1))

print("Ответ:", answer)

**Контрольный пример:**

number = 5, 2, 0;

**Результат выполнения программы:**

****

# Задание№18

Даны два целых числа A и B (при этом A ≤ B). Выведите все числа от A до B включительно.

**Входные данные:** a, b - int

**Выходные данные:** i - int

**Блок-схема:**



**Код:**

a = int(input("Введите число а: "))

b = int(input("Введите число б: "))

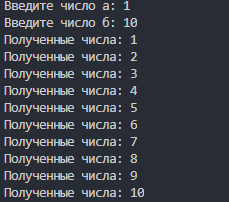
for i in range(a, b + 1):

    print("Полученные числа:",i)

**Контрольный пример:**

a = 1; b = 10;

**Результат выполнения программы:**

****

# Задание№19

Даны два целых числа A и В. Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае.

**Входные данные:** a, b - int

**Выходные данные:** i - int

**Блок-схема:**



**Код:**

a = int(input("Введите число а: "))

b = int(input("Введите число b: "))

if a < b:

    for i in range(a, b + 1):

        print("Полученные числа:",i)

else:

    for i in range(a, b - 1, -1):

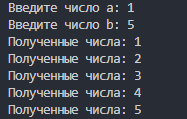
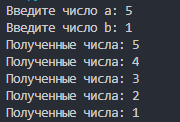
        print("Полученные числа:",i)

**Контрольный пример:**

a = 1; b = 5;

a = 5; b = 1;

**Результат выполнения программы:**

# Задание№20

Даны два целых числа A и В, A>B. Выведите все нечётные числа от A до B включительно, в порядке убывания. В этой задаче можно обойтись без инструкции if.

**Входные данные:** a, b -int

**Выходные данные:** i - int

**Блок-схема:**



**Код:**

a = int(input("Введите число а: "))

b = int(input("Введите число b: "))

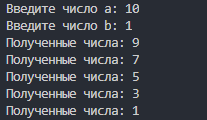
for i in range(a - (a + 1) % 2, b - b % 2, -2):

    print("Полученные числа:",i)

**Контрольный пример:**

a = 10; b = 1;

**Результат выполнения программы:**

****

# Задание№21

По данному натуральному n вычислите сумму 13+23+33+...+n3

**Входные данные:** number - int

**Математические операции:**

sum = sum + i^3

**Выходные данные:** sum - int

**Блок-схема:**

****

**Код:**

number = int(input("Введите число: "))

sum = 0

for i in range(1, number + 1):

    sum += i \*\* 3

print("Сумма кубов:",sum)

**Контрольный пример:**

number = 5;

**Результат выполнения программы:**

****

# Задание№22

Факториалом числа n называется произведение 1 × 2 × ... × n. Обозначение: n! По данному натуральному n вычислите значение n!. Пользоваться математической библиотекой math в этой задаче запрещено.

**Входные данные:** number - int

**Математические операции:**

result = 1

result = result \* i

**Выходные данные:** result - int

**Блок-схема:**



**Код:**

result = 1

number = int(input("Введите число: "))

for i in range(1, number + 1):

    result \*= i

print("Результат:",result)

**Контрольный пример:**

number = 10;

**Результат выполнения программы:**

****

# Задание№23

По данному натуральном nn вычислите сумму 1!+2!+3!+...+n!. В решении этой задачи можно использовать только один цикл. Пользоваться математической библиотекой math в этой задаче запрещено.

**Входные данные:** number - int

**Математические операции:**

partialFactorial = 1

partialSum = 0

partialFactorial \*= i

partialSum += partialFactorial

**Выходные данные:** partialSum - int

**Блок-схема:**

****

**Код:**

number = int(input("Введите число: "))

partialFactorial = 1

partialSum = 0

for i in range(1, number + 1):

    partialFactorial \*= i

    partialSum += partialFactorial

print("Результат:",partialSum)

**Контрольный пример:**

number = 6;

**Результат выполнения программы:**

****

# Задание№24

Дано N чисел: сначала вводится число N, затем вводится ровно N целых чисел. Подсчитайте количество нулей среди введенных чисел и выведите это количество.

**Входные данные:** repeat, number - int

**Математические операции:**

zeroes = 0

zeroes = zeroes + 1

**Выходные данные:** zeroes - int

**Блок-схема:**

****

**Код:**

zeroes = 0

repeat = int(input("Сколько раз выполнить программу: "))

for i in range(repeat):

    number = int(input("Введите число: "))

    if  number == 0:

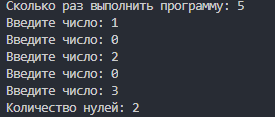
        zeroes += 1

print("Количество нулей:",zeroes)

**Контрольный пример:**

repeat = 5; number = 1, 0, 2, 0, 3;

**Результат выполнения программы:**

****

# Задание№25

По данному натуральному n ≤ 9 выведите лесенку из n ступенек, i-я ступенька состоит из чисел от 1 до i без пробелов

**Входные данные:** number - int

**Выходные данные:** j - int

**Блок-схема:**

****

**Код:**

number = int(input("Введите число: "))

for i in range(1, number + 1):

    for j in range(1, i + 1):

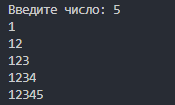
        print(j, sep='', end='')

    print()

**Контрольный пример:**

number = 5;

**Результат выполнения программы:**

****

# Задание№26

Для настольной игры используются карточки с номерами от 1 до N. Одна карточка потерялась. Найдите ее, зная номера оставшихся карточек. Дано число N, далее N − 1 номер оставшихся карточек (различные числа от 1 до N). Программа должна вывести номер потерянной карточки. Для самых умных: массивами и аналогичными структурами данных пользоваться нельзя.

**Входные данные:** cards, card - int

**Математические операции:**

card = 0

card = card + i

card = card - numTest

**Выходные данные:** card - int

**Блок-схема:**

****

**Код:**

cards = int(input("Введите количество карт: "))

card = 0

for i in range(1, cards + 1):

    card += i

for i in range(cards - 1):

    numTest = int(input("Введите число: "))

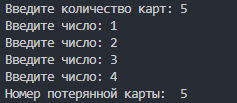
    card -= numTest

print("Номер потерянной карты: ",card)

**Контрольный пример:**

cards = 5; card = 1, 2, 3, 4;

**Результат выполнения программы:**

****