**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**(ФГБОУ ВО «АмГУ»)**

Факультет математики и информатики

Факультет математики и информатики

Кафедра информационных и управляющих систем

Направление подготовки 09.03.04 - Программная инженерия

Направленность (профиль) образовательной программы: Программная инженерия

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

на тему: **Работа с циклами**

|  |  |  |
| --- | --- | --- |
| Исполнитель  студент группы 357-об | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | В.Н. Корнышев |
| Проверил | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Е.В. Дегтярёв |

Благовещенск 2023

**1 ОБЩИЕ ЗАДАНИЯ**

**1 Задание.**

Последовательно вводится ненулевые числа. Определить сумму положительных и сумму отрицательных чисел. Закончить ввод чисел при вводе 0. Для перевода из строки в целое число, использовать класс int().

pos\_numb\_sum = 0  
neg\_numb\_sum = 0  
numb\_input=1  
  
while numb\_input != 0:  
 numb\_input = int(input('Введите число: '))  
 if numb\_input > 0 :  
 pos\_numb\_sum = pos\_numb\_sum + numb\_input  
 else:  
 neg\_numb\_sum = neg\_numb\_sum + numb\_input  
  
print ('Сумма положительных чисел: ', pos\_numb\_sum)  
print ('Сумма отрицательных чисел: ', neg\_numb\_sum)

Рисунок 1 – Листинг рабочей программы «2.1.py»

Таблица 1

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| -1  -3231  12321  43  0 | Сумма положительных чисел: 12364  Сумма отрицательных чисел: -3232 |
| 12  432  -11  0 | Сумма положительных чисел: 444  Сумма отрицательных чисел: -11 |
| 0 | Сумма положительных чисел: 0  Сумма отрицательных чисел: 0 |
| 3214  -2111  0 | Сумма положительных чисел: 3214  Сумма отрицательных чисел: -2111 |

**2 Задание.**

При помощи цикла распечатать ряд Фибоначчи: 1 1 2 3 5 8 13 21

fibo\_numbers = (1, 1, 2, 3, 5, 8, 13, 21)  
for n in fibo\_numbers:  
 print (n)

Рисунок 2 – Листинг рабочей программы «2.2.py»

Таблица 2

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Список чисел: 1,1,2,3,5,8,13,21 | Вывод ряда Фибоначчи  1  1  2  3  5  8  13  21 |

**3 Задание.**

Запрашивается 10 чисел (целые значения от 0 до 1000). Опишите алгоритм, позволяющий найти и вывести минимальное значение среди введенных чисел, которые имеют четное значение и не делятся на три.

numb\_list = []  
for numb in range(10):  
 numb\_list.append(int(input('Введите число: ')))  
numb\_list.sort()  
for numb in numb\_list:  
 if numb % 2 == 0 and numb % 3 != 0:  
 print('Ваше число: ', numb)  
 break  
else:  
 print('Такого числа введено не было.')

Рисунок 3 – Листинг рабочей программы «2.3.py»

Таблица 3

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  2  4  5  6  7  8  9  0  1 | Ваше число: 2 |
| 0  0  0  0  0  0  0  0  0  0 | Такого числа введено не было. |

**4 Задание.**

Составить программу для вычисления среднеарифметического N произвольных вводимых чисел.

sum\_of\_numb = 0  
N = 10  
  
for o in range(N):  
 number = int(input("Введите число: "))  
 sum\_of\_numb += number  
  
print('Ваш ответ: {0}'.format(sum\_of\_numb/10))

Рисунок 4 – Листинг рабочей программы «2.4.py»

Таблица 4

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 1  2  3  4  5  6  7  8  9  0 | Ваш ответ: 4.5 |

**5 Задание.**

Исправить предыдущее задание (2.4) для работы со случайными числами.

from random import randint  
  
sum\_of\_numb = 0  
N = 10  
  
for o in range(N):  
 number = randint(1,1000)  
 sum\_of\_numb += number  
  
print('Ваш ответ: {0}'.format(sum\_of\_numb/10))

Рисунок 5 – Листинг рабочей программы «2.5.py»

Таблица 5

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Список случайных чисел | Ваш ответ: 661.2 |

**6 Задание.**

Найдите все трехзначные и четырехзначные числа Армстронга.

Числом Армстронга считается натуральное число, сумма цифр которого, возведенных в N-ную степень (N – количество цифр в числе) равна самому числу. Например, 153 = 13 + 53 + 33

for numb in range (100,10000):  
 numb = str(numb)  
 list\_numb = sum([int (numb\_2) \*\* len (numb) for numb\_2 in numb])  
 if list\_numb == int(numb):  
 print('Вот это - ваше число Армстронга: ', numb)

Рисунок 6 – Листинг рабочей программы «2.6.py»

Таблица 6

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Числа от 100 до 10000 | Вот это - ваше число Армстронга: 153  Вот это - ваше число Армстронга: 370  Вот это - ваше число Армстронга: 371  Вот это - ваше число Армстронга: 407  Вот это - ваше число Армстронга: 1634  Вот это - ваше число Армстронга: 8208  Вот это - ваше число Армстронга: 9474 |

**7 Задание.**

Напишите программу, которая запрашивает натуральное число N и выводит на экран все автоморфные числа, не превосходящие N.

Автоморфным называется натуральное число, если оно равно последним цифрам своего квадрата. Например, 252 = 625

N = int(input('Введите натуральное число (N): '))  
if N>0:  
 for n1 in range (N+1):  
 sqr\_of\_n1 = n1 \*\* 2  
 if str(sqr\_of\_n1).endswith((str(n1))):  
 print('{0} - автоморфное число. Квадрат его равен {1}'.format(n1,sqr\_of\_n1))  
else:  
 print('Введено некорректное значение')

Рисунок 7 – Листинг рабочей программы «2.7.py»

Таблица 7

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 100 | 0 - автоморфное число. Квадрат его равен 0  1 - автоморфное число. Квадрат его равен 1  5 - автоморфное число. Квадрат его равен 25  6 - автоморфное число. Квадрат его равен 36  25 - автоморфное число. Квадрат его равен 625  76 - автоморфное число. Квадрат его равен 5776 |

**8 Задание.**

Распечатать дни недели с их порядковыми номерами. Кроме того, рядом выводить выходной это день, или рабочий

names\_list = ('Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье')  
for n in range (0,7):  
 if n<5:  
 print('{0} день - {1}. Это - рабочий день.'.format(n+1,names\_list[n]))  
 else:  
 print('{0} день - {1}. Это - выходной день.'.format(n + 1, names\_list[n]))

Рисунок 8 – Листинг рабочей программы «2.8.py»

Таблица 8

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
|  | 1 день - Понедельник. Это - рабочий день.  2 день - Вторник. Это - рабочий день.  3 день - Среда. Это - рабочий день.  4 день - Четверг. Это - рабочий день.  5 день - Пятница. Это - рабочий день.  6 день - Суббота. Это - выходной день.  7 день - Воскресенье. Это - выходной день. |

**2.ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**1 Задание.**

Вычислить предел последовательности an с точностью 0,5⸳10-6, если значения a0, x (x>1) вводятся. Значения an считать по правилу an = 5 – sin (1 – an-12)/xn.

import math  
  
a\_0 = float(input('Введите a(0): '))  
x = float(input('Введите x (x>1): '))  
  
if x <= 1:  
 print('Введено некорректное значение x. Перезапустите программу и повторите попытку.')  
else:  
 n = 1  
 a\_n\_prev = a\_0  
 while True:  
 a\_n = 5 - math.sin(1-a\_n\_prev\*\*2)/x\*\*n  
 n += 1  
 if abs(a\_n - a\_n\_prev) <= 0.5\*10\*\*(-6):  
 print("Предел последовательности равен: {0}".format(round(a\_n, 9)))  
 break  
 else:  
 a\_n\_prev = a\_n

Рисунок 9 – Листинг рабочей программы «2.9.py»

Таблица 9

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 2 2 | Предел последовательности равен: 4.999999568 |
| 5 10 | Предел последовательности равен: 4.999999991 |
| 0 100 | Предел последовательности равен: 5.0 |
| 1 1 | Введено некорректное значение x. Перезапустите программу и повторите попытку. |