**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**(ФГБОУ ВО «АмГУ»)**

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:Функции и процедуры

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

Благовещенск 2023

**1 ОБЩИЕ ЗАДАНИЯ**

**1 Задание.**

Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с последней.

def number\_col(number:float) -> None:

    """Процедура которая выводит на экран в столбик все цифры переданного ей числа (начиная с последней).

    Args:

        number (float): переданное число

    """

    if number == int(number):

        number = int(number)

    number = str(abs(number))[::-1]

    print("Цифры переданного числа: ")

    for i in number:

        if i == ".":

            continue

        print(i)

if \_\_name\_\_ == "\_\_main\_\_": # Если скрипт запущен как основной

    number\_col(float(input("Введите число: ")))

Рисунок 1 – Листинг рабочей программы «file1.py»

Таблица 1

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 123456 | Цифры переданного числа:  6  5  4  3  2  1 |
| -2.5 | Цифры переданного числа:  5  2 |
| -1234 | Цифры переданного числа:  4  3  2  1 |

**2 Задание.**

Напишите процедуру, которая выводит на экран все делители передаваемого ей числа (в одну строчку)

def divider(number:int) -> None:

    """Процедуру, которая определяет все делители передаваемого ей числа (в одну строчку)

    Args:

        number (int): передаваемое число

    """

    res = ""

    for i in range(1, abs(number)+1):

        if number % i == 0:

            res += f"{i} "

    if len(res) == 0:

        print("Таких чисел нет")

    print("Делители:", res)

if \_\_name\_\_ == "\_\_main\_\_": # Если скрипт запущен как основной

    divider(int(input("Введите число: ")))

Рисунок 2 – Листинг рабочей программы «file2.py»

Таблица 2

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 7 | Делители: 1 7 |
| -5 | Делители: 1 5 |
| 0 | Делители: Таких чисел нет |

**3 Задание.**

Составить программу с процедурой для вычисления степени числа (входные параметры: число и степень)

def in\_degree(number:int, degree:float) -> None:

    """Процедура, которая вычисляет степень числа

    Args:

        number (int): передаваемое число

        degree (float): передаваемая степень

    """

    if number < 0 and (degree \*\* (-1)) % 2 == 0:

        print("Невозможно из отрицательного числа вычислить четный корень")

    print("Число в степени", number \*\* degree)

if \_\_name\_\_ == "\_\_main\_\_": # Если скрипт запущен как основной

    in\_degree(int(input("Введите число: ")),

              float(input("Введите степень числа: ")))

Рисунок 3 – Листинг рабочей программы «file3.py»

Таблица 3

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 7  2 | 49 |
| -25  0.5 | Невозможно из отрицательного числа вычислить четный корень |
| 25  0.5 | 5.0 |

**4 Задание.**

Напишите процедуру, которая принимает параметр – натуральное число N – и выводит первые N чисел Фибоначчи.

def fibonacci(N:int) -> None:

    """Поиск и вывод чисел фибоначи

    Args:

        N (int): натуральное число

    """

    if N < 0:

        print("Число должно быть натуральным")

        return

    print("Ряд фибоначи:")

    if 0 <= N < 1:

        print(0)

        return

    n\_pas, n\_new = 1, 1

    for i in range(N):

        print(n\_pas)

        n\_new, n\_pas = n\_new + n\_pas, n\_new

if \_\_name\_\_ == "\_\_main\_\_": # Если скрипт запущен как основной

    fibonacci(int(input("Введите число N: ")))

Рисунок 4 – Листинг рабочей программы «file4.py»

Таблица 4

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 0 | Ряд фибоначи:  0 |
| 1 | Ряд фибоначи:  1 |
| 6 | Ряд фибоначи:  1  1  2  3  5  8 |
| -3 | Число должно быть натуральным |

**5 Задание.**

Напишите функцию, которая вычисляет количество цифр числа.

def len\_number(number:float) -> int:

    """Процедура, которая вычисляет количество цифр числа

    Args:

        number (float): передаваемое число

    Returns:

        int: количество цифр числа

    """

    number = abs(number) # Убираем знак числа

    k = 1 # Коофицент количества служебных знаков

    if number == int(number):

        k = 2

    return len(str(number)) - k

if \_\_name\_\_ == "\_\_main\_\_": # Если скрипт запущен как основной

    print(len\_number(float(input("Введите число: "))))

Рисунок 5 – Листинг рабочей программы «file5.py»

Таблица 5

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 123 | Количество цифр числа: 3 |
| -25 | Количество цифр числа: 2 |
| 0 | Количество цифр числа: 1 |
| 2.4 | Количество цифр числа: 2 |

**6 Задание.**

Напишите функцию, которая вычисляет факториал натурального числа N.

def factorial(N:int) -> int:

    """Функция, которая вычисляет факториал натурального числа  N

    Args:

        N (int): передаваемое число

    Returns:

        int: факториал числа

    """

    if N < 0:

        return "Факториал нельзя расчитать"

    if N in (0, 1):

        return 1

    fk = 1

    for i in range(1, N+1):

        fk \*= i

    return fk

if \_\_name\_\_ == "\_\_main\_\_": # Если скрипт запущен как основной

    print(factorial(int(input("Введите число: "))))

Рисунок 6 – Листинг рабочей программы «file6.py»

Таблица 6

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 0 | Факториал числа: 1 |
| -25 | Факториал числа: Факториал нельзя расчитать |
| 1 | Факториал числа: 1 |
| 6 | Факториал числа: 720 |

**2 ИНДИВИДУЛЬНОЕ ЗАДАНИЕ**

**2.4 Задание.**

Вводится последовательность из N целых положительных элементов. Посчитать количество чисел палиндромов. Для определения паллиндрома создать функцию

def found\_pall(number:str) -> bool:

    return number == number[::-1]

def main():

    k = 0

    for i in range(int(input("Введите количество элементов в последовательности: "))):

        number = input("Введите значение элемента: ")

        if found\_pall(number):

            k += 1

    print(f"Количество палиндромов: {k}")

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

    main()

Рисунок 7 – Листинг рабочей программы «main.py»

Таблица 7

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  101  111  121 | Количество палиндромов: 3 |
| 3  265  22  131 | Количество палиндромов: 2 |