**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 №6**

на тему: **Списки**

|  |  |  |
| --- | --- | --- |
| Исполнитель  студент группы 3105-об | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Д.Е. Буханов |
| Проверил | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Е.В. Дегтярёв |

Благовещенск 2023

**1 ОБЩИЕ ЗАДАНИЯ**

**1 Задание.**

Ввести строку, в которой записана сумма натуральных чисел,

например, ‘1+25+3’. Вычислите это выражение. Работать со строкой, как со

списком

def main():

    '''Функция main'''

    s = input('Введите строку с выражением: ')

    l = list(s)

    l += ["endl"]

    res = []

    ls\_command = ("+", "-", "endl")

    number\_op = ""

    for i in l:

        if i in ls\_command:

            res.append(int(number\_op))

            if i == "endl":

                res = sum(res)

            elif i == "+":

                number\_op = ""

            elif i == "-":

                number\_op = "-"

        else:

            number\_op += i

    print(f"Результат: {res}")

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

    main()

Рисунок 1 – Листинг рабочей программы «file1.py»

Таблица 1

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 105+76+56 | Результат: 237 |

**2 Задание.**

Дан список из 5 различных элементов. Используя функции (не использовать цикл), необходимо найти и вывести:

* минимальный и максимальный элементы списка;
* сумму и среднее арифметическое;
* второй минимальный элемент (второй по минимальности).

def main():

    '''Функция main'''

    lst = [4, 5, 2, 3, 4]

    lst.sort()

    print(lst)

    print(f"Минимальный элемент списка: {lst[0]}")

    print(f"Максимальный элемент списка: {lst[-1]}")

    print(f"Сумма элементов списка: {sum(lst)}")

    print(f"Среднее арифметическое элементов списка: {sum(lst) / len(lst)}")

    print(f"Второй минимальный (второй по минимальности) элемент списка: {lst[1]}")

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

    main()

Рисунок 2 – Листинг рабочей программы «file2.py»

Таблица 2

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Список находящийся в программе | Минимальный элемент списка: 2  Максимальный элемент списка: 5  Сумма элементов списка: 18  Среднее арифметическое элементов списка: 4.5  Второй минимальный (второй по минимальности) элемент списка: 3 |

**3 Задание.**

Проверить, является ли заданное слово палиндромом.

Примечание:

* Пример палиндрома: казак, ABBA
* Использовать функции.
* Поскольку при присваивании одного списка другому, изменение первого ведет к аналогичному изменению второго списка, то необходимо использовать копию (copy).

import copy

def check\_palid(st:str) -> bool:

    """Проверка на палидром

    Args:

        st (str): Строка для проверки

    """

    lst\_new = copy.deepcopy(st)

    lst\_new = lst\_new[::-1]

    return st == lst\_new

def main():

    '''Функция main'''

    stroka =  input('Введите слово: ')

    lst = list(stroka) # конвертируем строку в список

    if check\_palid(lst):

        print("Введенная строка - палиндром")

    else:

        print("Введенная строка - не палиндром")

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

    main()

Рисунок 3 – Листинг рабочей программы «file3.py»

Таблица 3

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| рор | Введенная строка - палиндром |

**4 Задание.**

Найдите в массиве все простые числа и скопируйте их в новый массив.

def check\_easy(number:int) -> bool:

    """Проверка на числа на простоту

    Args:

        number(int): Строка для проверки

    """

    sqrt\_number = abs(number) \*\* (1/2) # Находим корень для числа

    if sqrt\_number == int(sqrt\_number):  return False

    for i in range(2, int(sqrt\_number) + 1):

        if number % i == 0:  return False

    return True

def main():

    '''Функция main'''

    ls = input("Введите значения для массива (через ; ): ").split(";")

    ls\_easy = []

    for i in ls:

        if check\_easy(int(i)):  ls\_easy.append(i)

    print(f"Простые числа: {' '.join(ls\_easy)}")

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

    main()

Рисунок 4 – Листинг рабочей программы «file4.py»

Таблица 4

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 44;22;11;7;5;3 | Простые числа: 11 7 5 3 |

**5 Задание.**

Решить задачу поиска среднего значения в списке из N элементов. Использовать метод добавления элементов списка и суммирования элементов

def main():

    '''Функция main'''

    my\_list = []

    N = int(input("Введите количество элементов списка: "))

    for i in range(N):

        my\_list.append(float(input("Введите значение для элемента списка: ")))

    print(f"Среднее значение для списка {sum(my\_list) / N}")

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

    main()

Рисунок 5 – Листинг рабочей программы «file5.py»

Таблица 5

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 4  43  23  53  21 | Среднее значение для списка 35.0 |

**6 Задание.**

Определить индексы элементов массива (списка), значения которых принадлежат заданному диапазону (т.е. не меньше заданного минимума и не больше заданного максимума)

def main():

    '''Функция main'''

    my\_list = []

    while True:

        number = input("Введите значение для элемента списка (break - остановка ввода): ")

        if number == "break":  break

        my\_list.append(float(number))

    range\_min = float(input("Введите минимальное значение для диапазона: "))

    range\_max = float(input("Введите максимальное значение для диапазона: "))

    cout = int(input("Введите количество элементов: "))

    ls\_ind = []

    for i in range(len(my\_list)):

        if (range\_min <= my\_list[i] <= range\_max) and cout:

            ls\_ind.append(str(i))

            cout -= 1

        elif not cout:

            break

    print(f"Индексы элементов попадающие в диапазон: {' '.join(ls\_ind)}")

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

    main()

Рисунок 6 – Листинг рабочей программы «file6.py»

Таблица 6

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 76  32  45  67  23  break | Введите минимальное значение для диапазона: 40  Введите максимальное значение для диапазона: 70  Введите количество элементов: 2  Индексы элементов попадающие в диапазон: 2 3 |

**7 Задание.**

Дополнить предыдущую программу следующим: После того, как элемент с подходящим значением добавлен в новый список – удалять его из исходного списка.

def main():

    '''Функция main'''

    my\_list = []

    while True:

        number = input("Введите значение для элемента списка (break - остановка ввода): ")

        if number == "break":  break

        my\_list.append(float(number))

    range\_min = float(input("Введите минимальное значение для диапазона: "))

    range\_max = float(input("Введите максимальное значение для диапазона: "))

    cout = int(input("Введите количество элементов: "))

    ls\_ind = []

    for i in range(len(my\_list)):

        if (range\_min <= my\_list[i] <= range\_max) and cout:

            ls\_ind.append(str(i))

            my\_list[i] = False # Удаляем элемент из списка

            cout -= 1

        elif not cout:

            break

    print(f"Индексы элементов попадающие в диапазон: {' '.join(ls\_ind)}")

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

    main()

Рисунок 7 – Листинг рабочей программы «file7.py»

Таблица 7

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 76  32  45  67  3456  234  32  32  4  5  3  65  34  56  break | Введите минимальное значение для диапазона: 10  Введите максимальное значение для диапазона: 100  Введите количество элементов: 4  Индексы элементов попадающие в диапазон: 2 3 7 8 |