**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук

Кафедра информационных и управляющих систем

Направление подготовки 09.03.04 - Программная инженерия

Направленность (профиль) образовательной программы: Программная инженерия

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

на тему: **Массивы**

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

Благовещенск 2023

**1 ОБЩИЕ ЗАДАНИЯ**

**1 Задание.**

Дан массив. Необходимо подтвердить, что в массиве есть числа, кратные трем.

def main():

    '''Функция main'''

    ls = []

    while True:

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

        if number == "break":  break

        ls.append(float(number))

    for i in ls:

        if i % 3 == 0:

            print(f"Введенный маcсив: {ls}")

            print("В  массиве есть числа, кратные трем.")

            break

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

    main()

Листинг рабочей программы «file1.py»

Таблица 1

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3213  32131332  32342  break | Введенный масcив: [3213.0, 32131332.0, 32342.0]  В массиве есть числа, кратные трем. |

**2 Задание.**

Заполните массив случайными числами в диапазоне 0..4 и выведите на экран номера всех элементов, равных значению X (оно вводится с клавиатуры).

import random

def main():

    '''Функция main'''

    ls = []

    LEN = 20

    for i in range(LEN):

        ls.append(random.randint(0, 4))

    X = float(input("введите значение X: "))

    ls\_ind = []

    for i in range(LEN):

        if ls[i] == X:

            ls\_ind.append(str(i + 1))

    print(f"Введенный маcсив: {ls}")

    print(f"Номера элементов, равных значению X: {'Их нет' if len(ls\_ind) == 0 else ' '.join(ls\_ind) }")

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

    main()

Листинг рабочей программы «file2.py»

Таблица 2

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 2 | Введенный маcсив: [0, 1, 1, 0, 2, 4, 3, 1, 3, 4, 1, 0, 2, 2, 1, 4, 4, 1, 4, 2]  Номера элементов, равных значению X: 5 13 14 20 |

**3 Задание.**

Заполните массив случайными числами. Найдите номера первого минимального и последнего максимального элемента массива.

import random

def main():

    '''Функция main'''

    ls = []

    LEN = 20

    for i in range(LEN):

        ls.append(random.randint(0, 4))

    min\_item\_index = None

    max\_item\_index = None

    for i in range(len(ls) - 1):

        if min\_item\_index == None and ls[i] < ls[i+1]:

            min\_item\_index = i

        if ls[i] > ls[i+1]:

            max\_item\_index = i

    print(f"Массив: {ls}")

    print(f"Номер первого минимального элемента массива {min\_item\_index + 1}")

    print(f"Номер последнего максимального элемента массива {max\_item\_index + 1}")

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

    main()

Листинг рабочей программы «file3.py»

Таблица 3

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Данные вводятся рандомно | Массив: [0, 0, 2, 1, 2, 0, 0, 2, 1, 2, 4, 3, 3, 2, 1, 1, 0, 4, 3, 3]  Номер первого минимального элемента массива 2  Номер последнего максимального элемента массива 18 |

**4 Задание.**

Необходимо написать программу, в которой сортировка выполняется

«методом камня» – самый «тяжёлый» элемент опускается в конец массива.

def main():

    '''Функция main'''

    ls = []

    LEN = int(input("Введите длину массива: "))

    for i in range(LEN):

        ls.append(float(input("Введите значение элента: ")))

    print(f"Массив до сортировки: {ls}")

    for i in range(LEN):

        itr = LEN - i - 1

        for ii in range(itr):

            if ls[ii] > ls[ii + 1]:

                ls[ii + 1], ls[ii] = ls[ii], ls[ii + 1]

    print(f"Массив после сортировки: {ls}")

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

    main()

Листинг рабочей программы «file4.py»

Таблица 4

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 4  2  32  54  32 | Массив до сортировки: [2.0, 32.0, 54.0, 32.0]  Массив после сортировки: [2.0, 32.0, 32.0, 54.0] |

**5 Задание.**

Необходимо написать программу, которая сортирует массив (быстрой сортировкой) по возрастанию первой цифры числа.

def quick\_sort(A, nStart, nEnd) -> list:

    if nStart >= nEnd: return

    L = nStart; R = nEnd

    X = A[(L+R)//2]

    while L <= R:

        while A[L] < X: L += 1 # разделение

        while A[R] > X: R -= 1

        if L <= R:

            A[L], A[R] = A[R], A[L]

        L += 1; R -= 1

        quick\_sort( A, nStart, R ) # рекурсивные вызовы

        quick\_sort( A, L, nEnd )

def main():

    '''Функция main'''

    ls = []

    LEN = int(input("Введите длину массива: "))

    for i in range(LEN):

        ls.append(float(input("Введите значение элента: ")))

    print(f"Массив до сортировки: {ls}")

    quick\_sort(ls, 0, LEN - 1)

    print(f"Массив после сортировки: {ls}")

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

    main()

Листинг рабочей программы «file5.py»

Таблица 5

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  3  2  1 | Массив до сортировки: [3.0, 2.0, 1.0]  Массив после сортировки: [1.0, 2.0, 3.0] |

**6 Задание.**

Напишите программу, которая, не изменяя заданный массив, выводит номера его элементов в возрастающем порядке. Использовать вспомогательный массив номеров.

def main():

    '''Функция main'''

    ls = []

    LEN = int(input("Введите длину массива: "))

    for i in range(LEN):

        ls.append(float(input("Введите значение элента: ")))

    print(f"Введеный массив: {ls}")

    print(f"Номера элементов  в возрастающем порядке: {[ls.index(i) + 1 for i in sorted(ls)]}")

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

    main()

Листинг рабочей программы «file6.py»

Таблица 6

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  43  564  234 | Введеный массив: [43.0, 564.0, 234.0]  Номера элементов в возрастающем порядке: [1, 3, 2] |

**7 Задание.**

Напишите программу, которая сортирует массив и находит количество различных чисел в нём. Не использовать встроенные функции.

def my\_fun(arr):

    LEN = len(arr)

    sp\_set = []

    for i in range(LEN):

        itr = LEN - i - 1

        for ii in range(itr):

            if arr[ii] > arr[ii + 1]:

                arr[ii + 1], arr[ii] = arr[ii], arr[ii + 1]

        if arr[i] not in sp\_set:

            sp\_set.append(arr[i])

    return arr, len(sp\_set)

def main():

    '''Функция main'''

    ls = []

    LEN = int(input("Введите длину массива: "))

    for i in range(LEN):

        ls.append(float(input("Введите значение элента: ")))

    print(f"Введеный массив: {ls}")

    arr, sp\_set = my\_fun(ls)

    print(f"Отсортированный массив: {arr}")

    if sp\_set == 1:

        print(f"Количество различных чисел в массиве: {sp\_set}")

    else:

        print("Количество различных чисел в массиве: их нет")

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

    main()

Листинг рабочей программы «file7.py»

Таблица 7

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  443  234  235 | Введенный массив: [443.0, 234.0, 235.0]  Отсортированный массив: [234.0, 235.0, 443.0]  Количество различных чисел в массиве: 3 |

**8 Задание.**

Дан массив. Назовем серией группу подряд идущих одинаковых элементов, а длиной серии ‒ количество этих элементов. Сформировать два новых массива, в один из них записывать длины всех серий, а во второй — значения элементов, образующих эти серии.

def my\_fun(arr):

    res = []

    flag = [arr[0], 1]

    for i in arr[1:]:

        if i == flag[0]:

            flag[1] += 1

        else:

            res.append(flag)

            flag = [i, 1]

    res.append(flag)

    return res

def main():

    '''Функция main'''

    ls = []

    LEN = int(input("Введите длину массива: "))

    for i in range(LEN):

        ls.append(float(input("Введите значение элента: ")))

    print(f"Введеный массив: {ls}")

    res = my\_fun(ls)

    print(res)

    print(f"Значения элементов образующие серии: {[i[0] for i in res]}")

    print(f"Длины всех серий одинаковых элементов: {[i[1] for i in res]}")

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

    main()

Листинг рабочей программы «file8.py»

Таблица 8

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 5  23  23  4  3  3 | Введеный массив: [23.0, 23.0, 4.0, 3.0, 3.0]  [[23.0, 2], [4.0, 1], [3.0, 2]]  Значения элементов образующие серии: [23.0, 4.0, 3.0]  Длины всех серий одинаковых элементов: [2, 1, 2] |

**9 Задание.**

Напишите вариант метода пузырька, который заканчивает работу, если на очередном шаге внутреннего цикла не было перестановок. Не использовать встроенные функции.

def bubble\_sort(arr):

    LEN = len(arr)

    for i in range(LEN):

        itr = LEN - i - 1

        for ii in range(itr):

            if arr[ii] > arr[ii + 1]:

                arr[ii + 1], arr[ii] = arr[ii], arr[ii + 1]

            else: return

def main():

    ls = []

    LEN = int(input("Введите длину массива: "))

    for i in range(LEN):

        ls.append(float(input("Введите значение элента: ")))

    bubble\_sort(ls)

    print("Отсортированный массив:", ls)

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

    main()

Листинг рабочей программы «file9.py»

Таблица 9

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  2  4  3 | Отсортированный массив: [2.0, 3.0, 4.0] |