

***Кафедра информационных систем***

|  |  |
| --- | --- |
| ***Рейтинговая работа*** | ***Расчетно-аналитическое задание*** |
| ***по дисциплине*** | ***Алгоритмизация и программирование*** |
|  |  |
| ***Задание / Вариант №*** | ***Индивидуальное задание в соответствие с ФИО и ID*** |
|  |  |
| ***Тема*** | ***Алгоритмы типовых задач*** |
|  |  |
| ***Выполнена обучающимся группы*** | Серебряков Юрий Владимирович |
| ***ФИО обучающегося*** |  |
|  |  |
| ***Преподаватель*** | Зайцев Сергей Александрович |

Москва – 2021 г.

**Оглавление**

[Введение 3](#_Toc91030006)

[Постановка задачи 3](#_Toc91030007)

[Блок-схема 4](#_Toc91030008)

[Листинг программы 5](#_Toc91030009)

[Результат работы программы 8](#_Toc91030010)

[Вывод 8](#_Toc91030011)

[Список использованных источников 9](#_Toc91030012)

# Введение

Дан ряд целых чисел, полученный в результате обработки символов Фамилии, Имени, Отчества обучающегося, а также его ID. Необходимо отсортировать набор данных двумя способами, а также выполнить некоторые арифметические расчеты.

***Исходные данные***

ФИО и ID подготавливаются заранее в текстовом файле с именем source\_data.txt в кодировке Unicode (Юникод). Каждое слово, составляющее ФИО должно быть с заглавной буквы. Слова между собой должны быть разделены пробелом. Точка в конце не ставится. Текстовый файл с исходными данными должен иметь следующий вид (рисунок 1):

Серебряков Юрий Владимирович

70174464

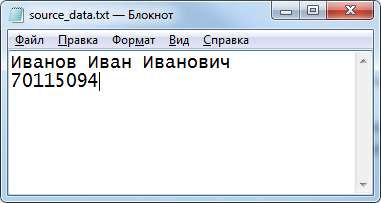


Рисунок 1 – Исходные данные

# Постановка задачи

В данной задаче требуется прочитать данные из одного файла, перевести ФИО в числа, значение которых будет ID юникода, преобразование будем делать с помощью функции ord(), отсортировать эти числа, сделать несколько операций и записать результат в другой файл.

Объем данных у нас небольшой, поэтому можно использовать простые алгоритмы сортировки, следовательно, выбираем метод пузырька. Также нам не нужна высокая точность сортировки, потому что мы не работаем со сложными дробями, комплексными числами или чем-то подобным, потому можно использовать алгоритм невысокой точности, значит нам также подойдет алгоритм быстрой сортировки.

# Блок-схема

Блок схема изображена на рисунке 2.

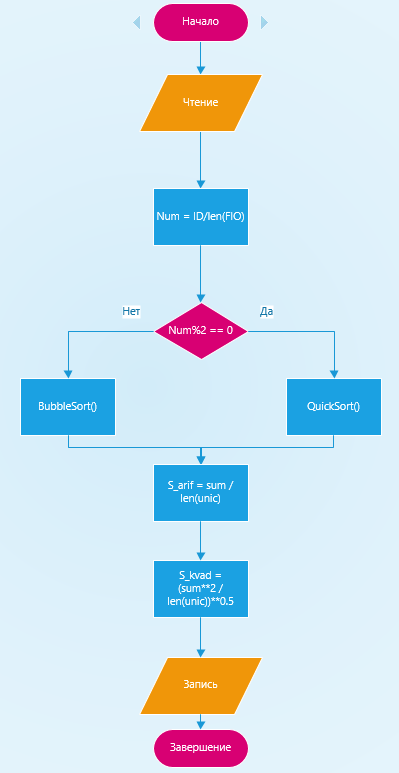


Рисунок 2 – блок схема

# Листинг программы

from os import replace

# Метод быстрой сортировки по возрастанию

def partition(unic, low, high):

    pivot = unic[(low + high) // 2]

    i = low - 1

    j = high + 1

    while True:

        i += 1

        while unic[i] < pivot:

            i += 1

        j -= 1

        while unic[j] > pivot:

            j -= 1

        if i >= j:

            return j

        unic[i], unic[j] = unic[j], unic[i]

def quick\_sort(unic):

    def \_quick\_sort(items, low, high):

        if low < high:

            split\_index = partition(items, low, high)

            \_quick\_sort(items, low, split\_index)

            \_quick\_sort(items, split\_index + 1, high)

    \_quick\_sort(unic, 0, len(unic) - 1)

#Метод пузырька по убыванию

def bubble\_sort(unic):

    swapped = True

    while swapped:

        swapped = False

        for i in range (len(unic) - 1):

            if (unic[i] < unic[i+1]):

                unic[i], unic[i+1] = unic[i+1], unic[i]

                swapped = True

#Чтение Файлов

source = open('source\_data.txt', encoding='utf-8')

res = open('result.txt', 'w', encoding='utf=8')

FIO = source.readline().strip()

ID = source.readline().strip()

# Запись в файл

res.write("1. Исходные данные: " + FIO + "; " + "ID: " + ID + "\n")

#Считаем число, на основе которого будет выбрано направление сортировки

num = int(int(ID) / len(FIO))

# Запись в файл

res.write("2. " + str(num) + "\n")

if (num % 2 == 0):

    direct = "по возрастанию"

    parity = "четное"

else:

    direct = "по убыванию"

    parity = "нечетное"

# Запись в файл

res.write("3. Направление сортировки: " + direct + ", так как число " + str(num) + " - " + parity + "\n")

#Перевод строки в ID юникода

unic = FIO.replace(" ", "")

unic = [ord(c) for c in unic]

# Запись в файл

res.write("4. Набор данных: " + str(unic) + "\n")

#Определяем направление сортировки и сортируем

if (num%2 == 0):

    quick\_sort(unic)

 # Запись в файл

    res.write("5. Отсортированный " + direct + " набор данных " + str(unic) + "\n")

else:

    bubble\_sort(unic)

# Запись в файл

    res.write("5. Отсортированный " + direct + " набор данных " + str(unic) + "\n")

# Складываем элементы списка

sum = unic[0]

for i in range (1,len(unic)):

    sum += unic[i]

#Ищем среднее арифметическое

s\_arif = round(sum / len(unic), 3)

# Запись в файл

res.write("6. Среднее арифметическое значение: " + str(s\_arif) + "\n")

#Склаыдваем квадраты элементов списка

sum = unic[0]\*\*2

for i in range (1,len(unic)):

    sum += unic[i]\*\*2

#ищем среднее квадратическое

s\_kvad = round((sum / len(unic))\*\*0.5, 3)

# Запись в файл

res.write("7. Среднее квадратическое значение: " + str(s\_kvad))

# Результат работы программы

1. Исходные данные: Серебряков Юрий Владимирович; ID: 70174464

2. 2506230

3. Направление сортировки: по возрастанию, так как число 2506230 - четное

4. Набор данных: [1057, 1077, 1088, 1077, 1073, 1088, 1103, 1082, 1086, 1074, 1070, 1088, 1080, 1081, 1042, 1083, 1072, 1076, 1080, 1084, 1080, 1088, 1086, 1074, 1080, 1095]

5. Отсортированный по возрастанию набор данных [1042, 1057, 1070, 1072, 1073, 1074, 1074, 1076, 1077, 1077, 1080, 1080, 1080, 1080, 1081, 1082, 1083, 1084, 1086, 1086, 1088, 1088, 1088, 1088, 1095, 1103]

6. Среднее арифметическое значение: 1079.385

7. Среднее квадратическое значение: 1079.445

# Вывод

Используя методы строк, алгоритмы сортировок, функции, методы чтения и записи в файл, мы считали данные с файла, сделали нужные преобразования и записали результат в нужном формате.

# Список использованных источников

1. ГОСТ 7.32-2017 СИБИД. Отчет о научно исследовательской работе. Структура и правила оформления
2. Лекции электронного университета МУИВ
3. Python world уроки программирования python для начинающих – URL: https://pythonworld.ru (Дата обращения 10.12.2021)
4. CodeRoad перевод популярных вопросов со StackOverflow – URL: <https://coderoad.ru> (Дата обращения 10.12.2021)