Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 2**

Дисциплина: Низкоуровневое программирование

Тема: Программирование EDSAC

Вариант №3

Выполнила

студентка гр. 3530901/10003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Коняшова

(подпись)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Коренев

(подпись)

“ ” 2022 г.

Санкт-Петербург

2022

**ОГЛАВЛЕНИЕ**

[1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc116413980)

[2. МЕТОД РЕШЕНИЯ 4](#_Toc116413982)

[3. ПРОГРАММА ORDERS1 5](#_Toc116413984)

[4. РАБОТА ПРОГРАММЫ ORDERS1 7](#_Toc116413985)

[5. ПРОГРАММА ORDERS2 8](#_Toc116413991)

[6. РАБОТА ПРОГРАММЫ ORDERS2 10](#_Toc116414088)

[7. ВЫВОД 10](#_Toc116414095)

1. **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

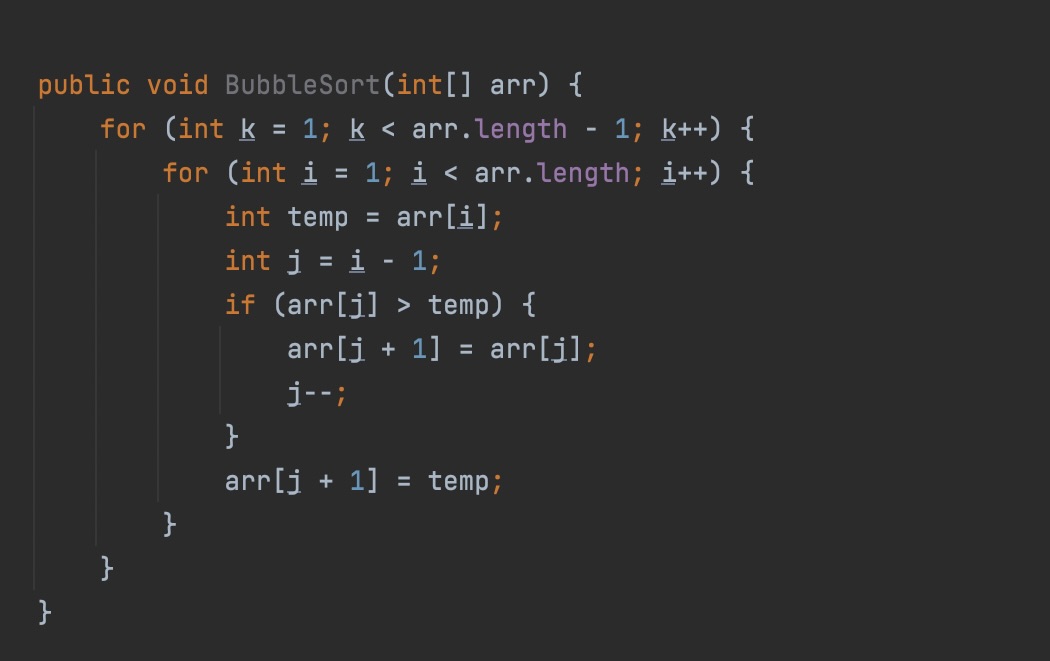
Разработать программу для EDSAC, реализующую сортировку обменом массива чисел in-place.

Программа должна реализовывать заданную функциональность и предполагает загрузчик Initials Orders 1. Массивы данных и другие параметры располагаются в памяти по фиксированным адресам.

Заданную функциональность необходимо выделить в замкнутую подпрограмму, разработать вызывающую её тестовую программу, используя возможности загрузчика Initial Orders 2. Адрес обрабатываемого массива данных и другие параметры передавать через ячейки памяти с фиксированными адресами.

1. **МЕТОД РЕШЕНИЯ**

Во время сортировки обменом, за каждый проход элементы сравниваются попарно, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются N-1 раз. Будем опираться на алгоритм, написанный на языке Java.



1. **ПРОГРАММА ORDERS1**

[Сортировка обменом массива чисел in-place. ]

[Овет - отсортированный массив на месте старого (ячейки 104-109)]

[31] T 110 S

[Цикл k]

[32] T 0 S [Обнуление acc]

[33] A 101 S [+1]

[34] T 98 S [i = 1]

[Цикл i]

[35] T 0 S [Обнуление acc]

[36] A 98 S [Берём i из ячейки 98 и записываем в acc]

[37] X 0 S

[38] A 102 S [Берём указатель на первый элемент массива и записываем в acc]

[39] L 0 L [Сдвиг аккумулятора на 1 разряд влево]

[40] A 100 S [Записываем команду A 0 S из ячейки 100 в аккумулятор]

[41] T 42 S [Берём предыдущую команду из аккумулятора и записываем в ячейку 42]

[42] A 0 S [Прочитали i-й элемент]

[43] T 1 S [Положили в рабочую ячейку (temp)]

[44] A 98 S [Берём i]

[45] S 101 S [Делаем i--]

[46] U 99 S [Кладём результат в j]

[if]

[47] X 0 S

[48] X 0 S

[49] A 102 S [Берём указатель на первый элемент массива]

[50] L 0 L [Сдвиг acc на 1 разряд влево]

[51] A 100 S [Добавляем команду A 0 S]

[52] T 53 S [Записываем результат в ячейку 53]

[53] A 0 S [Прочитали arr(j)]

[54] U 2 S [Запомнили arr(j)]

[55] S 1 S [Сравнили с temp]

[56] G 70 S [Если acc < 0 значит arr(j) < arr(i), следовательно идём в конец цикла]

[57] X 0 S

[58] T 0 S [Обнуление acc]

[59] A 102 S [Берём указатель на первый элемент массива]

[60] A 99 S [+j]

[61] A 101 S [+1]

[62] L 0 L [Сдвиг acc на 1 разряд влево]

[63] A 96 S [Добавляем команду T 0 S]

[64] T 66 S [Записываем в ячейку 66, обнулем acc]

[65] A 2 S [Кладём в acc arr(j)]

[66] T 0 S [arr(j+1) = arr(j), потом обнуляем]

[67] A 99 S [Берём j]

[68] S 101 S [-1]

[69] U 99 S [j--]

[Конец if]

[70] T 0 S [Обнуляем acc]

[71] A 99 S [Берём j]

[72] A 102 S [Добавляем указатель на 1 элемент массива]

[73] A 101 S [+1]

[74] L 0 L [Сдвиг acc на 1 разряд влево]

[75] A 96 S [Добавляем команду T 0 S]

[76] T 78 S [Записываем всё из acc в 78 ячейку, обнуляем acc]

[77] A 1 S [Берём temp]

[78] T 0 S [arr(j+1) = temp]

[79] A 98 S [Берём i]

[80] A 101 S [+1]

[81] U 98 S [i++]

[82] S 103 S [i - len]

[83] G 35 S [Если i < len то идём в начало цикла]

[Конец цикла i]

[84] A 97 S [Берём k]

[85] A 101 S [+1]

[86] U 97 S [k++]

[87] S 103 S [k - len]

[88] G 32 S [Если k < len то идём в начало цикла]

[Сортировка закончена]

[89] T 0 S [Обнуление acc]

[90] Z 0 S [Остановка машины]

[91] X 0 S

[92] X 0 S

[93] X 0 S

[94] X 0 S

[95] X 0 S

[96] T 0 S [Команда для чтения из acc]

[97] P 0 L [k]

[98] P 0 L [i]

[99] P 0 S [j]

[100] A 0 S [Команда для добавления в acc]

[101] P 0 L [1]

[102] P 52 S [Указатель на первый элем массива]

[103] P 3 S [Длина массива (len)]

[104] P 4 S [1-й элем = 8]

[105] P 1 S [2-й элем = 2]

[106] P 2 L [3-й элем = 5]

[107] P 4 L [4-й элем = 9]

[108] P 0 L [5-й элем = 1]

[109] P 1 L [6-й элем = 3]

1. **РАБОТА ПРОГРАММЫ ORDERS1**

****

****



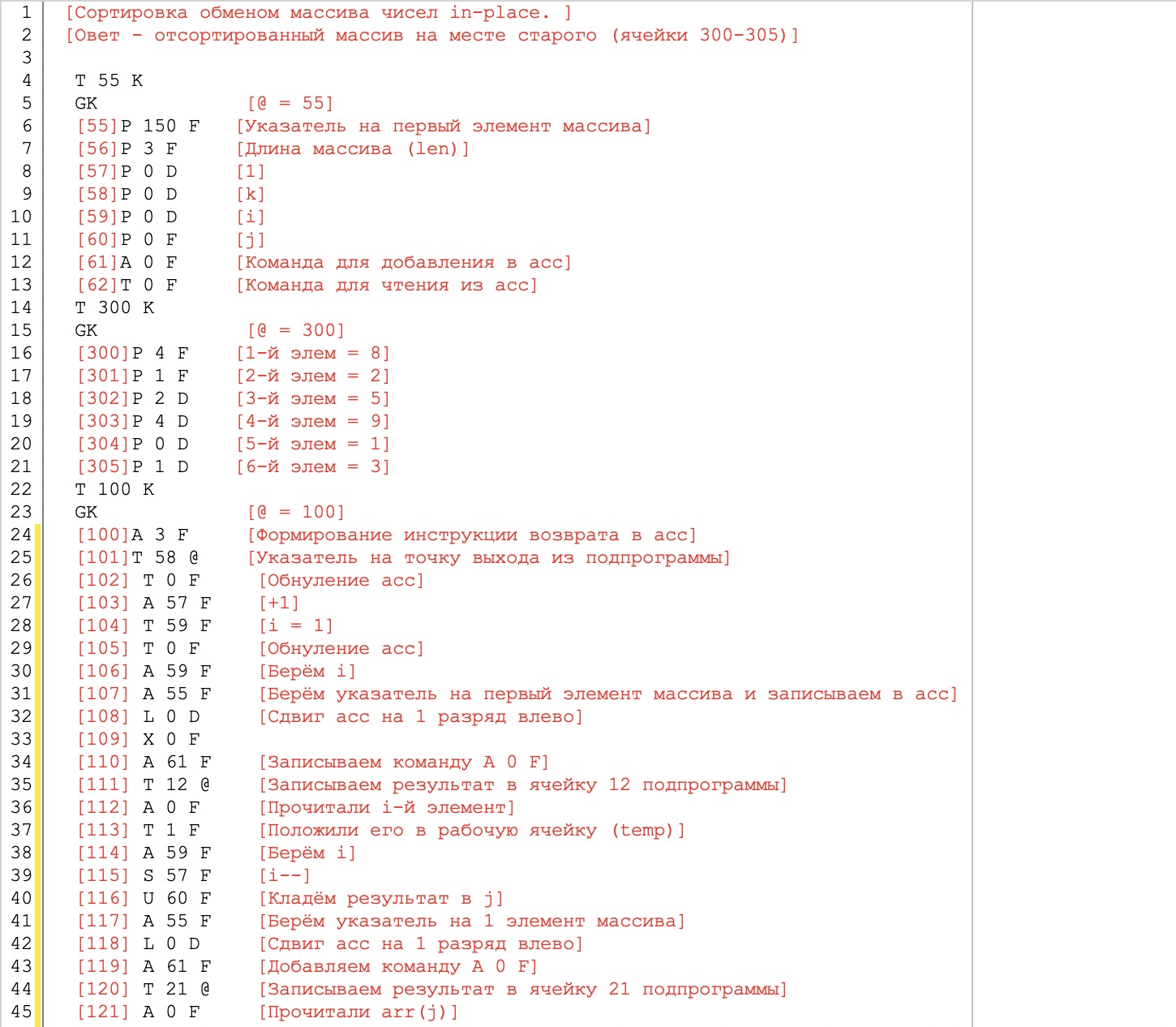


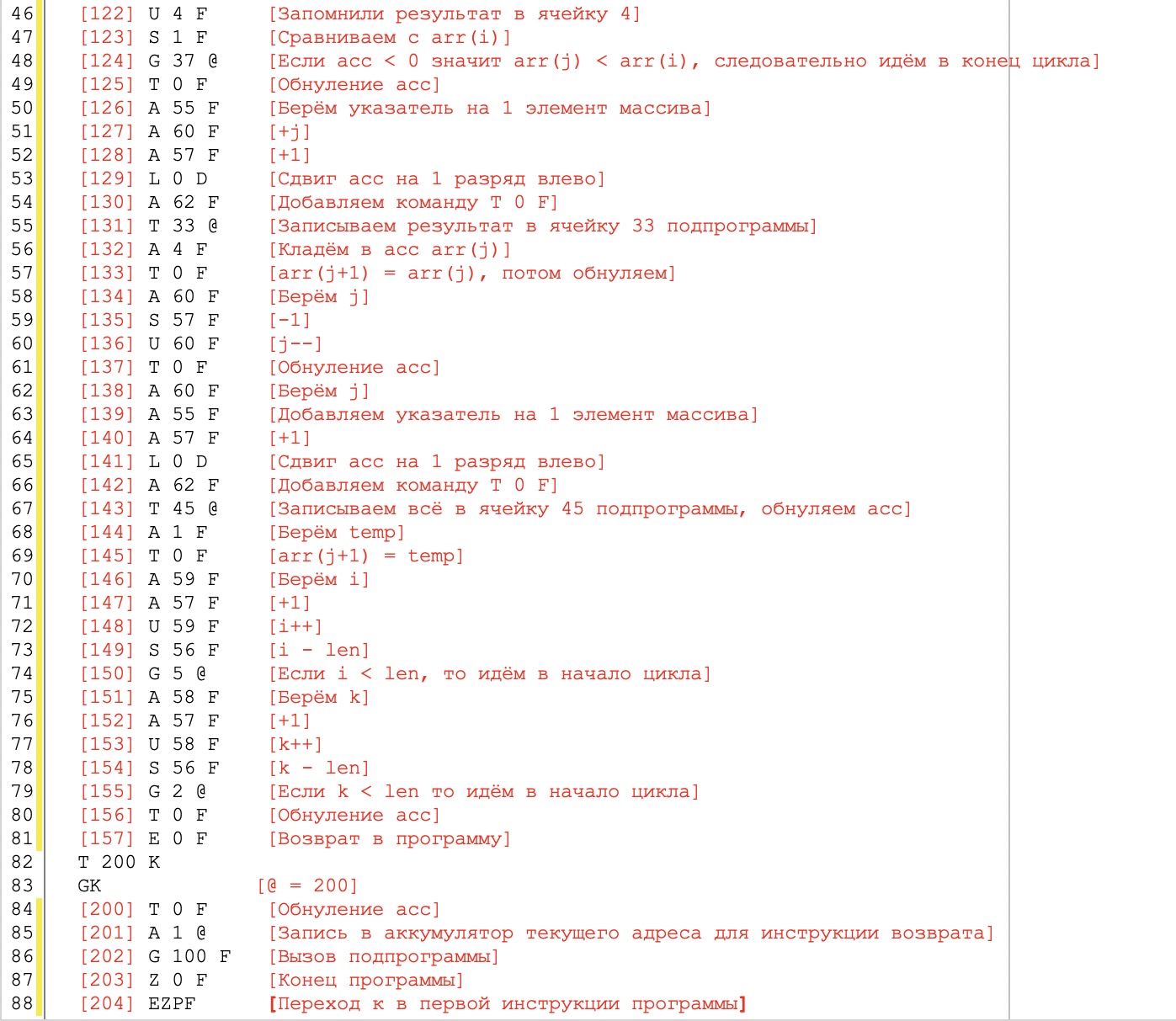




1. **ПРОГРАММА ORDERS2**

Основным отличием загрузчиков Initial Orders 2 от Initial Orders 1 является допустимость подпрограмм, где происходит относительная адресация ячеек, существенно упрощающая программирование. Абсолютная адресация обозначается буквой F, относительная @. Например, рассчитаем положения массива, который для удобство будем хранить за пределами программы с ячейки кратной 100. @ = 300, следовательно, первый элемент в ячейки 300, второй 300 + 1 = 301 и т.д.





1. **РАБОТА ПРОГРАММЫ ORDERS2**

****













1. **ВЫВОД**

В ходе выполнения данной лабораторной работы была разработана программа для EDSAC для пузырьковой сортировки массива.