Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Лабораторная работа № 2**

**EDSAC**

по дисциплине «Низкоуровневое программирование»

Выполнил

студент гр. 3530901/00004

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кручинин К. А.

(подпись)

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соболь. В.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Санкт-Петербург   
2021

**Задача**

В соответствии с условием 7 варианта требуется написать программу для EDSAC осуществляющую определение k-й порядковой статистики in-place.

Лабораторная работа делится на две части:

1. Разработать программу для EDSAC, реализующую определенную вариантом задания функциональность, и предполагающую загрузчик Initial Orders 1. Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.
2. Выделить определенную вариантом задания функциональность в замкнутую (closed) подпрограмму, разработать вызывающую ее тестовую программу. Использовать возможности загрузчика Initial Orders 2. Адрес обрабатываемого массива данных и другие параметры передавать через ячейки памяти с фиксированными адресами.

**Алгоритм**

Необходимо смоделировать программу для EDSAC, которая определит такой элемент неупорядоченного массива, если бы он был k-ым в упорядоченном. Для реализации сначала отсортируем массив произвольной длины алгоритмом сортировки вставками, опираясь на написанный на языке Kotlin алгоритм (рис. 1). Затем выведем k-ый элемент массива.

Изображение выглядит как текст

Автоматически созданное описание

Рис.1 Сортировка вставками на языке Kotlin.

**Программа для загрузчика Initial Orders 1.**

Изображение выглядит как стол

Автоматически созданное описание

Рис. 2 Программа для загрузчика IO1 строки 31–67.

**Изображение выглядит как текст

Автоматически созданное описание**

Рис. 3 Программа для загрузчика IO1 строки 68–110.

Выполним запуск программы. Массив чисел представлен на рисунке 4.

**Изображение выглядит как текст

Автоматически созданное описание**

Рис. 4 Исходный массив.

Результаты работы программы можно видеть на рисунках 5–13.

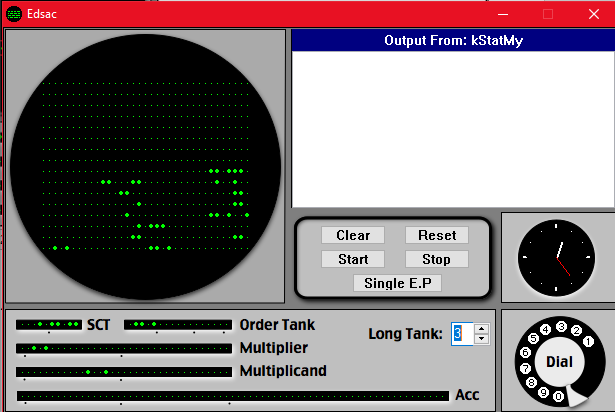


Рис. 5 Массив данных в симуляторе после выполнения программы.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Рис. 6-11 Каждый элемент массива данных после выполнения программы.

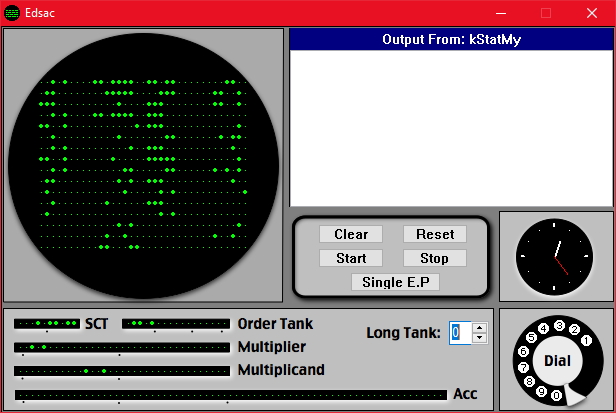




Рис. 12 и 13 Результат выполнения программы.

**Вывод по IO1**

Результаты выполнения программы на загрузчике IO1 полностью совпадают с ожидаемыми.

**Построение программы для загрузчика Initial Orders 2**

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 @. Также немного меняется синтаксис: S → F, L → D.

Изображение выглядит как стол

Автоматически созданное описание

Рис. 14 программа для загрузчика IO2 (1).

Изображение выглядит как стол

Автоматически созданное описание

Рис. 15 программа для загрузчика IO2 (2).

**Изображение выглядит как текст

Автоматически созданное описание**Рис. 16 Исходный массив.

Результаты работы программы видно на рисунках 17–25.

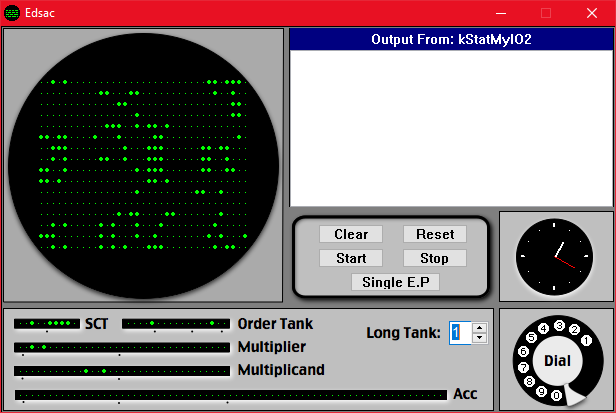


Рис. 17 Массив данных после выполнения программы

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Рис. 18-23 Каждый элемент массива данных после выполнения работы.

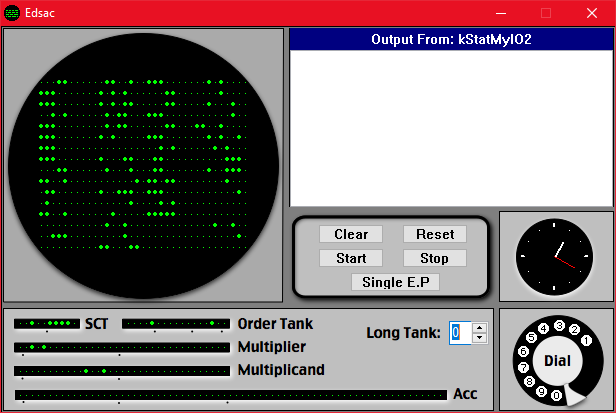




Рис. 24 и 25 Результаты работы программы.

**Вывод**

В ходе данной работы был осуществлён алгоритм сортировки вставками и вывод k-ого элемента массива в двух загрузчиках машины EDSAC. Результаты полностью соответствуют ожидаемым.