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. 3530901/90004

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шомов М.Ю.

(подпись)

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексюк А.О.

(подпись)

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

Санкт-Петербург   
2021

**Задача**

В соответствии с условием 19 варианта требуется написать программу для EDSAC осуществляющую циклический сдвиг массива.

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**

В связи с особенностями машины EDSAC требуется более тщательное планирование программы с учётом «чтения/записи» числа из ячейки из аккумулятора.

Потенциальная подпрограмма IO2

Начало

Число сдвигов – число элементов массива > 0?

да

Отнять число элементов массива

нет

Прибавить число элементов массива

Конец

Все сдвиги выполнены?

нет да

Внести последний элемент в память

Вернуть «каретку» в исходное положение, на последнюю и предпоследнюю позицию

Все переносы выполнены?

нет да

Переставить предыдущий элемент на место текущего

Сдвинуть «каретку» на один в начало

Потенциальная подпрограмма IO2

**\*«каретка»** – условное наименование пары команды, выполняющих чтение предыдущей ячейки и последующую запись в следующую.

**Построение программы для загрузчика Initial Orders 1**

1. T 166 S
2. T 0 S
3. A 146 S [чтение числа перестановок]
4. S 145 S [вычитание из числа перестановок числа элементов]
5. E 34 S [если результат положительный - повторение]
6. A 145 S [прибавление числа элементов - выведение в положительный диапазон]
7. T 146 S [запись нового числа сдвигов - без лишних полных оборотов]
8. ZS ["пауза"]
9. A 56 S [чтение изначального варианта полуфункции команды каретки]
10. T 147 S [сохранение команды]
11. A 57 S [чтение изначального варианта полуфункции команды каретки]
12. T 148 S [сохранение команды]
13. A 146 S [НЦ сдвигов, чтение числа сдвигов]
14. S 136 S [вычитание 1]
15. G 150 S [если сдвиги выполнены - завершение]
16. T 146 S [запись числа сдвигов]
17. A 164 S [чтение последнего элемента массива]
18. T 138 S [сохранение последнего элемента массива]
19. A 145 S [берём в акк число элементов, равное числу внутренних переносов]
20. S 136 S [вычитание 1]
21. T 149 S [сохранение числа переносов]
22. A 149 S [НЦ внутренних переносов]
23. S 136 S [вычитание 1]
24. G 65 S [если все переносы выполнены, то переход к постановке сохранённого последнего элемента в начало]
25. T 149 S [сохранение числа переносов]
26. A 163 S [функция "каретки" чтения предыдущего элемента]
27. T 164 S [функция "каретки" записи предыдущего элемента на место нынешнего]
28. A 56 S [чтение полукоманды "каретки"]
29. S 137 S [сдвиг "каретки"]
30. T 56 S [запись новой команды "каретки"]
31. A 57 S [чтение полукоманды "каретки"]
32. S 137 S [сдвиг "каретки"]
33. T 57 S [запись новой команды "каретки"]
34. E 52 S [КЦ внутренних переносов, возврат в начало цикла]
35. TS
36. A 138 S [финальное перемещение последнего элемента на место первого, чтение]
37. T 160 S [запись на первую позицию]
38. A 147 S [чтение изначальной команды "каретки"]
39. T 56 S [возврат изначальной версии]
40. A 148 S [чтение изначальной команды "каретки"]
41. T 57 S [возврат изначальной версии]
42. E 43 S [КЦ сдвигов, возврат в начало цикла]
43. X0S

…

1. X0S
2. P0L [1]
3. P1S [2]
4. X0S [хранение последнего элемента]
5. X0S
6. X0S
7. X0S
8. X0S
9. X0S
10. X0S
11. P2L [количество элементов = 5]
12. P2S [количество сдвигов = 4]
13. X0S [хранение изначальной версии полуфункции "каретки"]
14. X0S [хранение изначальной версии полуфункции "каретки"]
15. X0S [кол-во переносов внутри массива]
16. X0S
17. X0S
18. X0S
19. X0S
20. X0S
21. X0S
22. X0S
23. X0S
24. X0S
25. X0S
26. P1S [2]
27. P2S [4]
28. P3S [6]
29. P4S [8]
30. P5S [10]
31. Z0S

Выполняется запуск программы. Массив чисел 2, 4, 6, 8, 10 представлен рисунке 1.

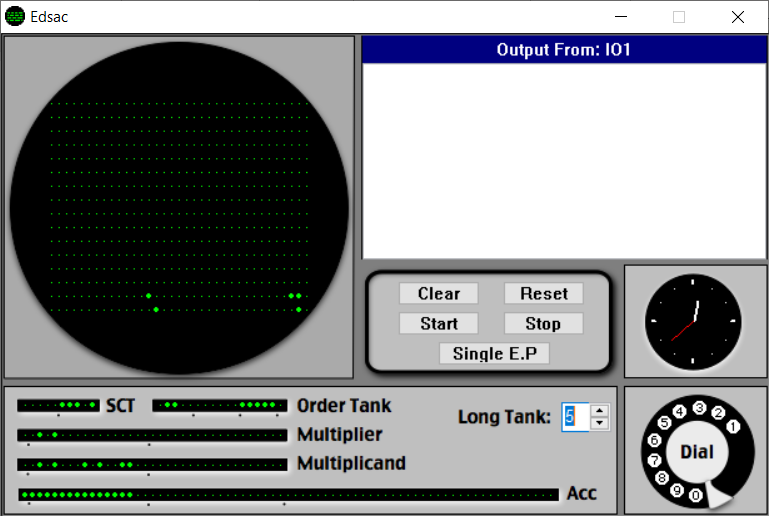


Рисунок 1 Массив чисел

После полного выполнения инструкций для сдвига 4 раза получили последовательность 4, 6, 8, 10, 2. Снимок экрана симулятора представлен рисунком 2.

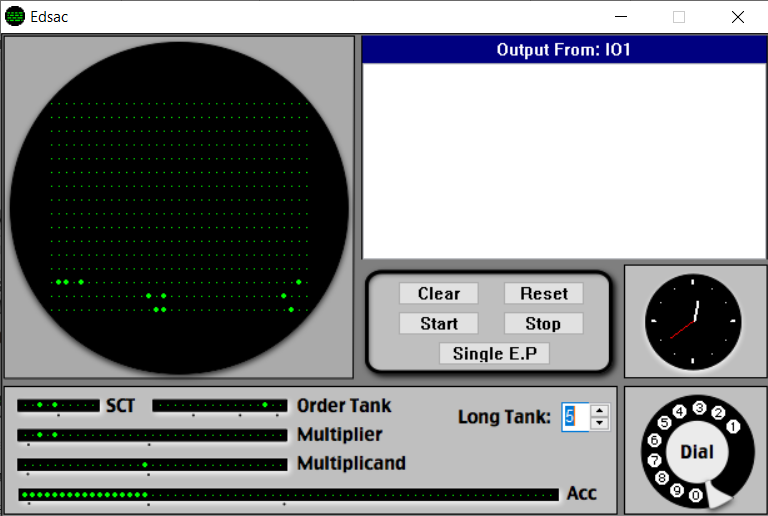


Рисунок 2 Массив чисел

Работоспособность программы на загрузчике Initial Orders 1 подтверждена.

**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.

T 56 K

P 0 D [1]

P 1 F [2]

XF

XF

XF

XF

XF

P 2 D [количество элементов = 5]

P 2 F [количество сдвигов = 4]

XF [кол-во внутр сдвигов]

T 256 K

P1F [2]

P2F [4]

P3F [6]

P4F [8]

P5F [10]

T 150 K [подпрограмма удаления лишних кругов сдвига]

GK

1. A 3 F [инструкция возврата]
2. T 8 @ [запись инструкции в конец подпрограммы]
3. A 64 F [чтение числа перестановок]
4. S 63 F [вычитание из числа перестановок числа элементов]
5. E 3 @ [если результат положительный - повторение]
6. A 63 F [прибавление числа элементов - выведение в положительный диапазон]
7. T 64 F [запись нового числа сдвигов - без лишних полных оборотов]
8. X 0 F
9. E 0 F

T 200 K [подпрограмма сдвигов]

GK

1. A 3 F [Инструкция возврата (формируется в аккумуляторе)]
2. T 37 @ [Записываем инструкцию возврата в конец]
3. A 19 @ [чтение изначального варианта полуфункции команды каретки]
4. T 38 @ [сохранение команды]
5. A 20 @ [чтение изначального варианта полуфункции команды каретки]
6. T 39 @ [сохранение команды]
7. A 64 F [НЦ сдвигов, чтение числа сдвигов]
8. S 56 F [вычитание 1]
9. G 36 @ [если сдвиги выполнены - завершение]
10. T 64 F [запись числа сдвигов]
11. A 260 F [чтение последнего элемента массива]
12. T 41 @ [сохранение последнего элемента массива]
13. A 63 F [берём в акк число элементов, равное числу внутренних переносов]
14. S 56 F [вычитание 1]
15. T 40 @ [сохранение числа переносов]
16. A 40 @ [НЦ внутренних переносов]
17. S 56 F [вычитание 1]
18. G 27 @ [если все переносы выполнены, то переход к постановке сохранённого последнего элемента в начало]
19. T 40 @ [сохранение числа переносов]
20. A 259 F [функция "каретки" чтения предыдущего элемента]
21. T 260 F [функция "каретки" записи предыдущего элемента на место нынешнего]
22. A 19 @ [чтение полукоманды "каретки"]
23. S 57 F [сдвиг "каретки"]
24. T 19 @ [запись новой команды "каретки"]
25. A 20 @ [чтение полукоманды "каретки"]
26. S 57 F [сдвиг "каретки"]
27. T 20 @ [запись новой команды "каретки"]
28. E 15 @ [КЦ внутренних переносов, возврат в начало цикла]
29. TF
30. A 41 @ [финальное перемещение последнего элемента на место первого, чтение]
31. T 256 F [запись на первую позицию]
32. A 38 @ [чтение изначальной команды "каретки"]
33. T 19 @ [возврат изначальной версии]
34. A 39 @ [чтение изначальной команды "каретки"]
35. T 20 @ [возврат изначальной версии]
36. E 6 @ [КЦ сдвигов, возврат в начало цикла]
37. TF
38. E 0 F

P0F [хранение изначальной версии полуфункции "каретки"]

P0F [хранение изначальной версии полуфункции "каретки"]

P0F [кол-во переносов внутри массива]

P0F [хранение последнего элемента]

T 100 K [программа]

GK

Z 0 F

T 0 F

A 2 @

G 150 F [вызов подпрограммы удаления кругов]

Z 0 F

T0F

A 2 @

G 200 F [вызов подпрограммы сдвигов]

Z 0 F

EZPF

Выполняется запуск программы. Массив чисел 2, 4, 6, 8, 10 представлен рисунке 3.

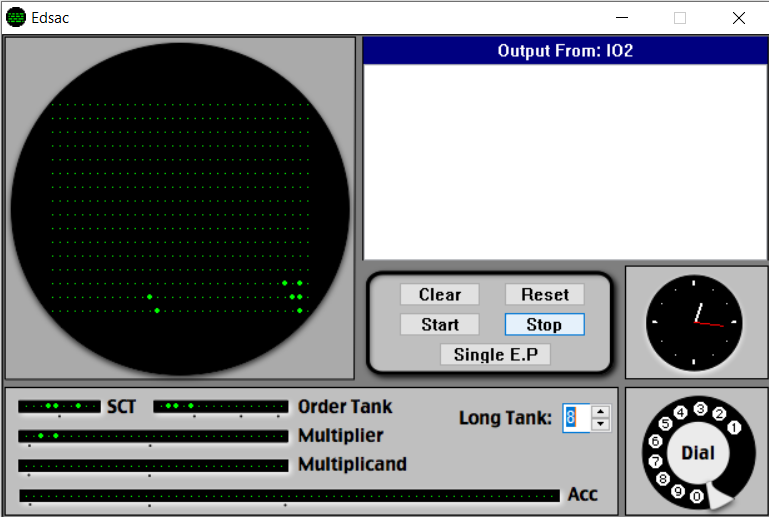


Рисунок 3 Массив чисел

После полного выполнения инструкций для сдвига 4 раза получили последовательность 4, 6, 8, 10, 2. Снимок экрана симулятора представлен рисунком 4.

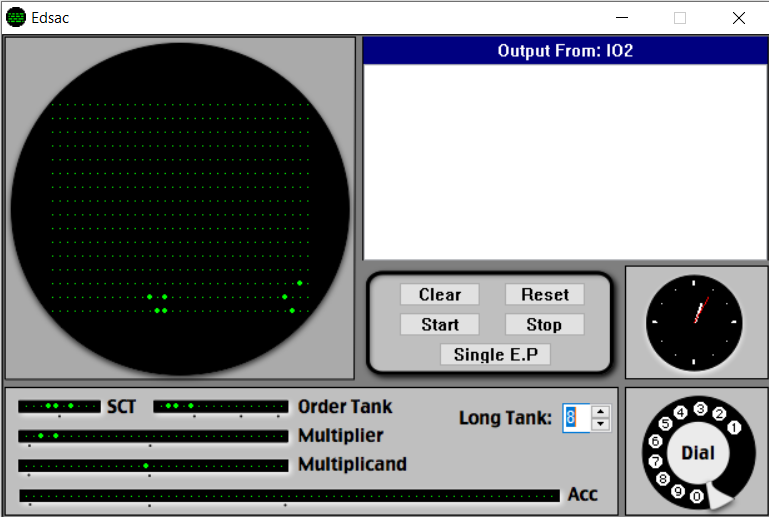


Рисунок 4 Массив чисел

Работоспособность программы на загрузчике Initial Orders 2 подтверждена.

**Вывод**

В ходе данной работы был осуществлён алгоритм циклического сдвига элементов массива в двух загрузчиках машины EDSAC. Результаты полностью соответствуют ожидаемым.