Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина**: Низкоуровневое программирование

**Тема**: Программирование EDSAC

Вариант: 1

Выполнил студент гр. 3530901/90005 Д. Е. Бакин

(подпись)

Преподаватель Д. С. Степанов

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

# Задачи работы

1. Разработать программу для EDSAC, реализующую определенную 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 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.

# Ход Работы

Функциональность, определенная вариантом задания:  
Формирование в памяти десятичного представления целого числа со знаком.

## Initial Order 1

Выполняя задачу работы, был написан код.

Листинг Код для Initial Order 1 представлен в приложении 1

В ходе выполнения данной программы заданное число преобразуется в символьное представление в памяти. Число задается в адресе 90. После выполнения программы число содержится в диапазоне адресов 100-105, где:

* 100 - знаковый разряд;
* 101 - 5 числовой разряд;
* 102 - 4 числовой разряд;
* 103 - 3 числовой разряд;
* 104 - 2 числовой разряд;
* 105 - 1 числовой разряд.

Пример:

На вход программы подается исходное число 228

На выходе получаем:

* 100 – значение 0 (соответствует плюсу);
* 101 – значение 0;
* 102 – значение 0;
* 103 – значение 2;
* 104 – значение 2;
* 105 – значение 8,

Что соответствует требуемой функциональности.

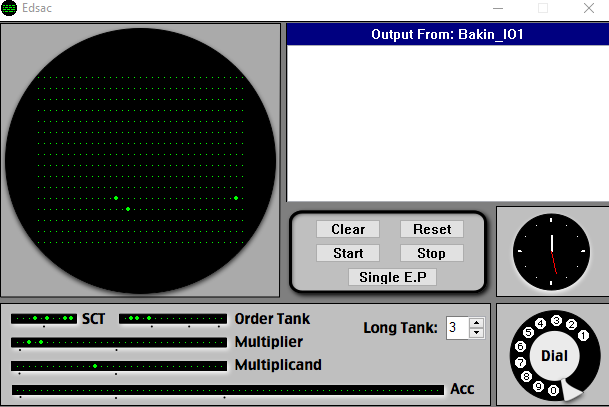


Рис. 1 Результат работы программы. Initial Order 1.

## Initial Order 2

Выполняя вторую задачу этой лабораторной работы, был написан следующий код.

Листинг Закрытая подпрограмма Initial Order 2 представлена в приложении 2

В данной программе исходное число задается в ячейку 1. Далее идет вызов подпрограммы, которая записывает исходное число в соответствующие ячейки памяти. После выполнения программы число содержится в диапазоне адресов 250-255, где:

250 - знаковый разряд;

251 - 5 числовой разряд;

252 - 4 числовой разряд;

253 - 3 числовой разряд;

254 - 2 числовой разряд;

255 - 1 числовой разряд.

Пример:

На вход программы подается исходное число -1337 (минимально возможное число в одной ячейке памяти).

На выходе получаем:

* 250 – значение 1 (соответствует минусу);
* 251 – значение 0;
* 252 – значение 1;
* 253 – значение 3;
* 254 – значение 3;
* 255 – значение 7,

Что соответствует требуемой функциональности.

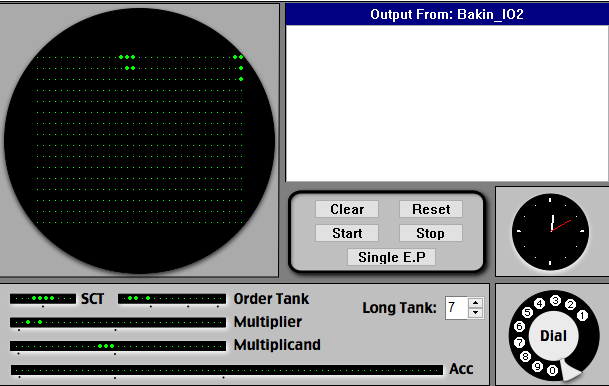


Рис. 2 Результат работы программы. Initial Order 2.

# Вывод

В данной работе мы ознакомились с работой EDSAC. Была разработана программа для EDSAC, используя загрузчик Initial Orders 1, реализующая определенную функциональность. Также, используя загрузчик Initial Orders 2, была выделена заданная функциональность в замкнутую подпрограмму и разработана вызывающая ее программа. Ожидаемые результаты совпали с результатами работы симулятора, программа работает верно.

**Код для Initial *Order 1***

T 91 S [31 - для IO1]

A 90 S [32 - добавление ячейки с исходным числом в аккумулятор 90 - ячейка с исходным числом]

E 38 S [33 - Если знак в аккумуляторе положительный перейти в позицию проверки старшего разрада, где 38 - адрес проверки старшего разряда]

[IF]

T 250 S [34 - Очистка аккумулятора]

A 84 S [35 - Добавление в аккумулятор константы со знаком минус 84 - константа со знаком минус]

T 100 S [36 - Перемещаю содержимое аккумулятора в ячейку памяти 100. Знаковый разряд]

S 90 S [37 - записываю в аккум модуль числа]

U 90 S [38 - Переношу содержимое аккумулятора в ячейку 90]

S 85 S [39 - вычитаю из аккумулятора константу 10тыс, где 85 - константа 10тыс]

G 47 S [40 - если знак аккумулятора отрицательный то переход в ячейку 47]

[IF]

T 90 S [41 - Переношу содержимое аккумулятора в ячейку 90]

A 101 S [42 - Переношу содержимое 5 разряда в аккумулятор]

A 89 S [43 - Прибавляю к аккумулятору константу 1 из ячейки 89]

T 101 S [44 - Переношу содержимое аккумулятора в 5 разряд]

A 90 S [45 - Возвращаю число после вычитания в аккумулятор]

E 38 S [46 - Перехожу к проверке 5 разряда]

T 0 S [47 - очистка аккумулятора]

A 90 S [48 - возвращение состояния числа до вычитания]

U 90 S [49 - Переношу содержимое аккумулятора в ячейку 90]

S 86 S [50 - вычитаю из аккумулятора константу 1тыс где 86 - константа 1тыс]

G 58 S [51 - если знак аккумулятора отрицательный, то перехожу в ячейку 58]

[IF]

T 90 S [52 - Переношу содержимое аккумулятора в ячейку 90]

A 102 S [53 - Переношу содержимое 4 разряда в аккумулятор]

A 89 S [54 - Прибавляю к аккумулятору константу 1 из ячейки 89]

T 102 S [55 - Переношу содержимое аккумулятора в 4 разряд]

A 90 S [56 - Возвращаю число после вычитания в аккумулятор]

E 49 S [57 - Перехожу к проверке 4 разряда]

T 0 S [58 - очистка аккумулятора]

A 90 S [59 - возвращение состояния числа до вычитания]

U 90 S [60 - Переношу содержимое аккумулятора в ячейку 150]

S 87 S [61 - вычитаю из аккумулятора константу 100, где 87 - константа 100]

G 69 S [62 - если знак аккумулятора отрицательный, то перехожу в ячейку 69]

[IF]

T 90 S [63 - Переношу содержимое аккумулятора в ячейку 90]

A 103 S [64 - Переношу содержимое 3 разряда в аккумулятор]

A 89 S [65 - Прибавляю к аккумулятору константу 1 из ячейки 89]

T 103 S [66 - Переношу содержимое аккумулятора в 3 разряд]

A 90 S [67 - Возвращаю число после вычитания в аккумулятор]

E 60 S [68 - Перехожу к проверке 3 разряда]

T 0 S [69 - очистка аккумулятора]

A 90 S [70 - возвращение состояния числа до вычитания]

U 90 S [71 - Переношу содержимое аккумулятора в ячейку 90]

S 88 S [72 - вычитаю из аккумулятора константу 10, где 88 - константа 10]

G 80 S [73 - если знак аккумулятора отрицательный, то перехожу в ячейку 80]

[IF]

T 90 S [74 - Переношу содержимое аккумулятора в ячейку 90]

A 104 S [75 - Переношу содержимое 2 разряда в аккумулятор]

A 89 S [76 - Прибавляю к аккумулятору константу 1 из ячейки 89]

T 104 S [77 - Переношу содержимое аккумулятора в 2 разряд]

A 90 S [78 - Возвращаю число после вычитания в аккумулятор]

E 71 S [79 - Перехожу к проверке 2 разряда]

T 0 S [80 - очистка аккумулятора]

A 90 S [81 - возвращение состояния числа до вычитания]

T 105 S [82 - Переношу значение 1 разряда в ячейку 105]

Z S [83 - Стоп для отладки]

P 0 L [84 - константа минус]

P 5000 S [85 - константа 10000]

P 500 S [86 - константа 1000]

P 50 S [87 - константа 100]

P 5 S [88 - константа 10]

P 0 L [89 - константа 1]

P 114 S [90 - Исходное число]

**Код для Initial *Order 2***

T 56 K [- директива IO2 установка адреса загрузки]

G K [- директива IO2 фиксация начального адреса подпрограмы]

[0:] A 3 F [- формирование кода инструкции возврата в аккумуляторе]

[1:] T 53 @ [- запись инструкции возврата]

[2:] A 1 F [- добавление ячейки с исходным числом в аккумулятор, 1 - ячейка с исходным числом]

[3:] E 8 @ [- Если знак в аккумуляторе положительный, перейти в позицию проверки старшего разрада, 8 - адрес проверки старшего разряда]

[IF]

[4:] T 300 F [- Очистка аккумулятора]

[5:] A 54 @ [- Добавление в аккумулятор константы со знаком минус, 54 - константа со знаком минус]

[6:] T 250 F [- Перемещаю содержимое аккумулятора в ячейку памяти 250 (знаковый разряд)]

[7:] S 1 F [- превращаю отрицательное число в положительное]

[ENDIF]

[8:] U 1 F [- Переносим содержимое аккумулятора в ячейку 1

[9:] S 55 @ [- вычитаю из аккумулятора константу 10000, где 55 - константа 10000]

[10:] G 17 @ [- если знак аккумулятора отрицательный переходим в ячейку 17]

[IF]

[11:] T 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[12:] A 251 F [- Переносим содержимое старшего разряда в аккумулятор]

[13:] A 59 @ [- Прибавляем к аккумулятору константу 1 из ячейки 59]

[14:] T 251 F [- Переносим содержимое аккумулятора в старший разряд]

[15:] A 1 F [- Возвращаем число после вычитания в аккумулятор]

[16:] E 8 @ [- Переходим к проверке старшего разряда]

[ENDIF]

[17:] T 300 F [- очистка аккумулятора]

[18:] A 1 F [- возвращаем состояния числа до вычитания]

[19:] U 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[20:] S 56 @ [- вычитаем из аккумулятора константу 1000, где 56 - константа 1000]

[21:] G 28 @ [- если знак аккумулятора отрицательный то переходим в ячейку 28]

[IF]

[22:] T 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[23:] A 252 F [- Переносим содержимое 3 разряда в аккумулятор]

[24:] A 59 @ [- Прибавляем к аккумулятору константу 1 из ячейки 59]

[25:] T 252 F [- Переносим содержимое аккумулятора в 3 разряд]

[26:] A 1 F [- Возвращаем число после вычитания в аккумулятор]

[27:] E 19 @ [- Переходим к проверке 3 разряда]

[ENDIF]

[28:] T 300 F [- очистка аккумулятора]

[29:] A 1 F [- возвращение состояния числа до вычитания]

[30:] U 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[31:] S 57 @ [- вычитаю из аккумулятора константу 100, где 57 - константа 100]

[32:] G 39 @ [- если знак аккумулятора отрицательный, то перехожу в ячейку 39]

[IF]

[33:] T 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[34:] A 253 F [- Переносим содержимое 2 разряда в аккумулятор]

[35:] A 59 @ [- Прибавляем к аккумулятору константу 1 из ячейки 59]

[36:] T 253 F [- Переносим содержимое аккумулятора в 2 разряд]

[37:] A 1 F [- Возвращаем число после вычитания в аккумулятор]

[38:] E 30 @ [- Переходим к проверке 2 разряда]

[ENDIF]

[39:] T 300 F [- очистка аккумулятора]

[40:] A 1 F [- возвращение состояния числа до вычитания]

[41:] U 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[42:] S 58 @ [- вычитаю из аккумулятора константу 10, где 58 - константа 10]

[43:] G 50 @ [- если знак аккумулятора отрицательный, то перехожу в ячейку 50]

[IF]

[44:] T 1 F [- Переносим содержимое аккумулятора в ячейку 1]

[45:] A 254 F [- Переносим содержимое 1 разряда в аккумулятор]

[46:] A 59 @ [- Прибавляем к аккумулятору константу 1 из ячейки 59]

[47:] T 254 F [- Переносим содержимое аккумулятора в 2 разряд]

[48:] A 1 F [- Возвращаем число после вычитания в аккумулятор]

[49:] E 41 @ [- Переходим к проверке 1 разряда]

[ENDIF]

[50:] T 300 F [- очистка аккумулятора]

[51:] A 1 F [- возвращение состояния числа до вычитания]

[52:] T 255 F [- Переношу значение младшего разряда в ячейку 255

[53:] E 0 F [эпилог: инструкция возврата из подпрограммы]

[54:] P 0 D [- константа минус]

[55:] P 5000 F [- константа 10000]

[56:] P 500 F [- константа 1000]

[57:] P 50 F [- константа 100]

[58:] P 5 F [- константа 10]

[59:] P 0 D [- константа 1]

**Программа, использующая подпрограмму.**

G K [- директива IO2, фиксация начального адреса программы]

[0:] A 5 @ [- загрузка в аккумулятор исходного числа]

[1:] T 1 F [- запись исходного числа в ячейку 1]

[2:] A 2 @ [- вызов]

[3:] G 56 F [- подпрограммы]

[4:] Z F [- остановка]

[5:] P 64867 D [- исходное число -1337]

EZ PF