

Отчёт по лабораторной работе

Дисциплина: Низкоуровневое программирование

Тема: EDSAC

Выполнил студент гр. 3530901/10005

_____ Кузичева П.Д.
(подпись)

Преподаватель

_____ Коренев Д. А.
(подпись)

“ ____ ” _____ 2022 г.

Санкт-Петербург

2022

Оглавление

ТЗ	3
Метод решения.....	3
Программа orders 1.....	3
Пояснение к программе.....	3
Код программы.....	4
Результат работы	5
Программа orders 2.....	5
Пояснение к программе.....	5
Код программы.....	6
Результат работы	7

ТЗ

Написать программу для машины EDSAC, которая реализует реверс массива. Необходимо реализовать эту программу для обоих загрузчиков: Initial Orders 1 и 2.

Метод решения

При реверсе массива первый элемент становится последним, а последний первым; второй — предпоследним, а предпоследний — вторым; третий элемент уходит на место третьего с конца, а тот на место третьего и т. д. Реверс массива осуществляется перестановкой этих пар элементов. Перестановок надо сделать в два раза меньше, чем длина массива.

Например, рассмотрим массив [1, 2, 3, 4]: в первую очередь будут переставлены первый и последний элемент. В данном случае 1 и 4. Получится массив [4, 2, 3, 1]. Следующими элементами для перестановки будут 2 и 3. После данной операции получится массив [4, 3, 2, 1], который мы и ожидали получить. Перестановок было выполнено две, а длина массива четыре.

Программа orders 1

Пояснение к программе

Программа начинается с 31-й ячейки, так как все предыдущие ячейки используются для работы загрузчика, но после начала программы их также можно использовать. Ячейки с 67-й по 76-ю используются для хранения вспомогательных значений:

- Ячейка 67 – инструкция для забора числа из нужной ячейки;
- Ячейка 68 – инструкция для добавления числа в нужную ячейку;
- Ячейка 69 – адрес первого элемента массива;
- Ячейка 70 – адрес последнего элемента массива;
- Ячейка 71 – счетчик итераций (количество необходимых перестановок пар);
- Ячейка 72 – константа 1 для итераций;
- Ячейка 73 – нулевой элемент массива, равный 1;
- Ячейка 74 – первый элемент массива, равный 2;
- Ячейка 75 – второй элемент массива, равный 3;
- Ячейка 76 – третий элемент массива, равный 5.

Сначала программа осуществляет загрузку счетчика итераций для того, чтобы посмотреть, сколько пар элементов надо еще поменять местами, в случае, если счетчик равен нулю или отрицателен, программа завершает свою работу. Далее происходит перемена местами последнего и первого элементов с загрузкой в аккумулятор их адресов из вспомогательной ячейки. При перестановке происходит перезапись инструкций. При этом используется команда L0L (сдвиг значения аккумулятора на один разряд влево) — это нужно по той причине, что поле адреса инструкции в EDSAC начинается не с 0-го разряда, а с 1-го. В конце программы происходит обновление адресов первого и последнего элементов массива с последующей записью обновленных значений на те же места. К адресу первого прибавляется единица, а из адреса последнего единица вычитается. Далее если счетчик итераций больше нуля, начинается новая итерация цикла.

Код программы

```

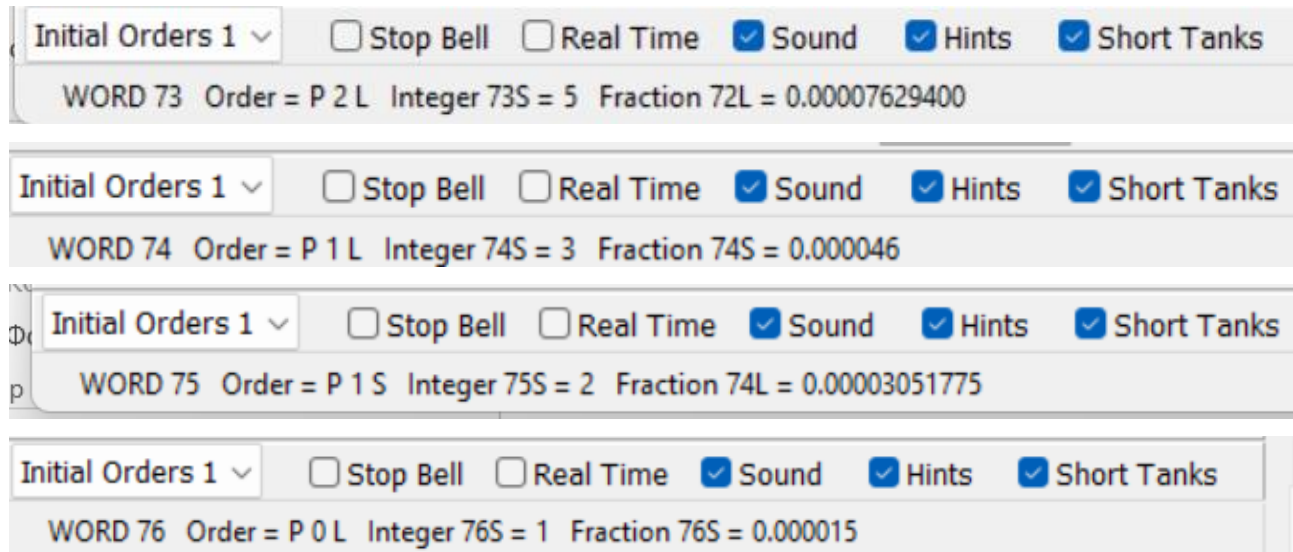
30
31 T [END:] 77 S [начало программы]
32 [LOOP: начало цикла] A [LEN:] 71 S [загрузка счетчика итераций]
33 S [CONST1:] 72 S [вычитание из счетчика константы 1]
34 G [END:] 66 S [если счетчик итераций меньше либо равен нулю, завершение програм]
35 T [LEN:] 71 S [загрузка обновленного значения счетчика итераций]
36 A [ADDR_START:] 69 S [загрузка в аккумулятор первого элемента пары]
37 L O L [сдвиг аккумулятора на один разряд влево]
38 A [TAKE:] 67 S [прибавляем код инструкции с нулевым полем адреса]
39 T [R1] 52 S [запись сформированной инструкции, обнуление аккумулятора]
40 A [ADDR_START:] 69 S [загрузка в аккумулятор первого элемента пары]
41 L O L [сдвиг аккумулятора на один разряд влево]
42 A [PUT:] 68 S [прибавляем код инструкции с нулевым полем адреса]
43 T [W1] 55 S [запись сформированной инструкции, обнуление аккумулятора]
44 A [ADDR_END:] 70 S [загрузка в аккумулятор второго элемента пары]
45 L O L [сдвиг аккумулятора на один разряд влево]
46 A [TAKE:] 67 S [прибавляем код инструкции с нулевым полем адреса]
47 T [R2] 54 S [запись сформированной инструкции, обнуление аккумулятора]
48 A [ADDR_END:] 70 S [загрузка в аккумулятор второго элемента пары]
49 L O L [сдвиг аккумулятора на один разряд влево]
50 A [PUT:] 68 S [прибавляем код инструкции с нулевым полем адреса]
51 T [W2] 57 S [запись сформированной инструкции, обнуление аккумулятора]
52 [R1:] A 0 S [загрузка в аккумулятор первого элемента из пары]
53 T 1 S [запись первого элемента из пары во вспомогательную ячейку 1]
54 [R2:] A 0 S [загрузка в аккумулятор второго элемента из пары]
55 [W1:] T 0 S [добавление первого элемента из пары]
56 A 1 S [взятие значение элемента из вспомогательной ячейки 1]
57 [W2:] T 0 S [добавление второго элемента из пары]
58 T 1 S [обнуление вспомогательной ячейки]
59 A [ADDR_START:] 69 S [добавление в аккумулятор адреса первого элемента из пары]
60 A [CONST1:] 72 S [прибавление единицы к нему]
61 T [ADDR_START:] 69 S [запись измененного адреса в ту же ячейку]
62 A [ADDR_END:] 70 S [добавление в аккумулятор адреса второго элемента из пары]
63 S [CONST1:] 72 S [вычитание из него единицы]
64 T [ADDR_END:] 70 S [запись измененного адреса в ту же ячейку]
65 E [LOOP_START:] 32 S [возвращение в начало цикла]
66 [END] Z 0 S [остановка программы]
67 [TAKE:] A 0 S [инструкция для забора числа из нужной ячейки]
68 [PUT:] T 0 S [инструкция для добавления числа в нужную ячейку]
69 [ADDR_START:] P 36 L [адрес первого элемента]
70 [ADDR_END:] P 38 S [адрес второго элемента]
71 [LEN:] P 1 S [счетчик итераций(количество необходимых перестановок пар)]
72 [CONST1:] P 0 L [константа 1 для итераций]
73 P 0 L [нулевой элемент массива, равный 1]
74 P 1 S [первый элемент массива, равный 2]
75 P 1 L [второй элемент массива, равный 3]
76 P 2 L [третий элемент массива, равный 5]

```

Результат работы

Исходный массив: 1(ячейка 73), 2(ячейка 74), 3(ячейка 75), 5(ячейка 76),

Ожидаемый результат: 5(ячейка 73), 3(ячейка 74), 2(ячейка 75), 1(ячейка 76).



Результат совпал с ожидаемым.

Программа orders 2

Пояснение к программе

При написании кода для Initial Orders 2 программа делится на две части: подпрограмма и программа, её вызывающая. В Initial Orders 2 используется относительная адресация — это значит, что адреса ячеек отсчитываются не с первой строчки программы, а с начального адреса подпрограммы. В программе номера ячеек, где хранятся результаты, совпадают фактическими, так как присутствует отступ, и текст подпрограммы начинается с 56 строчки.

Логика работы программы, набор констант и зарезервированных значений остаются точно такими же, как и в Initial Orders 1, за исключением решения избавиться от счетчика итераций. В данном случае его роль играет длина массива, из которой после каждой перестановки пары будет вычитаться константа 2. Также присутствуют изменения, которые присущи Initial Orders 2. Так, для обращения к ячейке или записи значения в неё по локальной адресации используется код длины слова “@”, а для относительной адресации — «F». Также в Initial Orders 2 другой формат записи констант: D вместо L и F вместо S.

Код программы

```

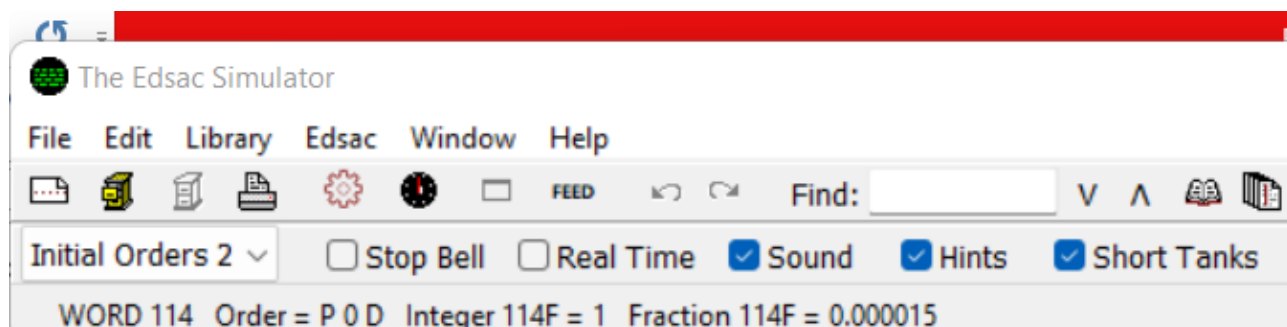
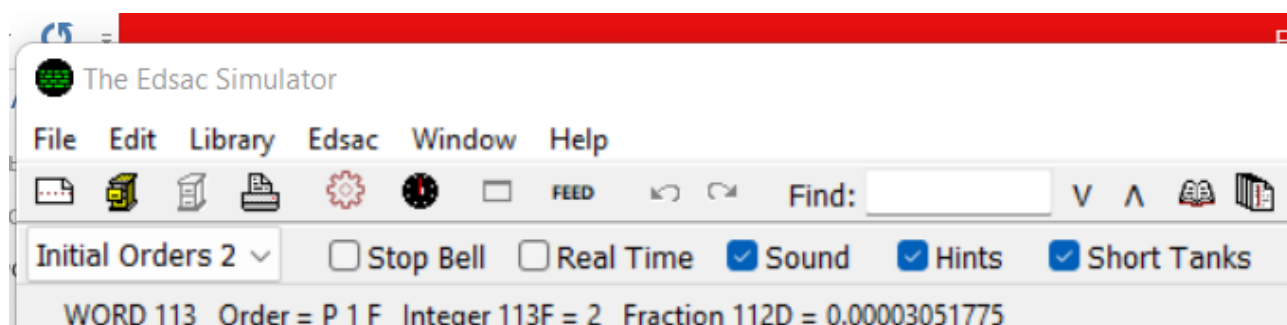
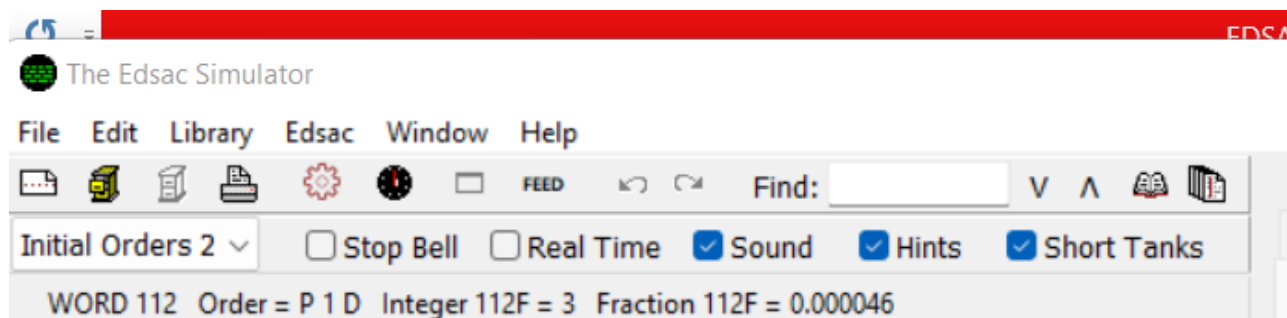
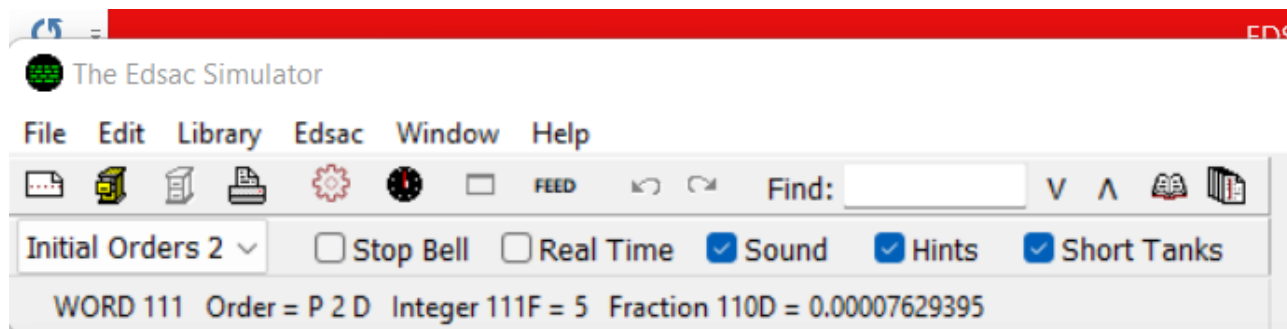
54 T 56 K [директива IO2, установка адреса загрузки]
55 G K [директива IO2, фиксация начального адреса подпрограммы]
56 [0] A 3 F [пролог: формирование инструкции возврата в Acc]
57 [1] T 37[RETURN] @ [пролог: запись инструкции возврата]
58 [2] [LOOP] A 2 [ADDR_START:] F [загрузка в аккумулятор адреса первого элемента пары]
59 [3] L 0 D [сдвиг аккумулятора на один разряд влево]
60 [4] A 38 [TAKE:] @ [прибавка к аккумулятору инструкции с нулевым адресом]
61 [5] T 22 [R1:] @ [запись сформированной инструкции в рабочую ячейку]
62 [6] A 2 [ADDR_START:] F [загрузка в аккумулятор адреса первого элемента пары]
63 [7] L 0 D [сдвиг аккумулятора на один разряд влево]
64 [8] A 39 [PUT:] @ [прибавка к аккумулятору инструкции с нулевым адресом]
65 [9] T 25 [W1:] @ [запись сформированной инструкции в рабочую ячейку]
66 [10] A 4 [ADDR_END:] F [загрузка в аккумулятор адреса второго элемента пары]
67 [11] L 0 D [сдвиг аккумулятора на один разряд влево]
68 [12] A 38 [TAKE:] @ [прибавка к аккумулятору инструкции с нулевым адресом]
69 [13] T 24 [R2:] @ [запись сформированной инструкции в рабочую ячейку]
70 [14] A 4 [ADDR_END:] F [загрузка в аккумулятор адреса второго элемента пары]
71 [15] L 0 D [сдвиг аккумулятора на один разряд влево]
72 [16] A 39 [PUT:] @ [прибавка к аккумулятору инструкции с нулевым адресом]
73 [17] T 27 [W2:] @ [запись сформированной инструкции в рабочую ячейку]
74 [18] A 0 [LEN:] F [загрузка счётчика итераций(длина массива)]
75 [19] S 41 [CONST1] @ [уменьшение счётчика на два(учет пары уже обработанных элементов)]
76 [20] G 36 [RETURN] @ [если счётчик меньше нуля либо равен нулю, завершение программы]
77 [21] T 0 [LEN:] F [загрузка обновленного значения счетчика итераций]
78 [22] [R1:] A 0 F [загрузка первого элемента из пары]
79 [23] T 1 F [запись первого элемента из пары во вспомогательную ячейку]
80 [24] [R2:] A 0 F [загрузка второго элемента из пары]
81 [25] [W1:] T 0 F [добавление первого элемента пары]
82 [26] A 1 F [взятие значение элемента из вспомогательной ячейки]
83 [27] [W2:] T 0 F [добавление второго элемента пары]
84 [28] T 1 F [обнуление вспомогательной ячейки]
85 [29] A 2 [ADDR_START:] F [загрузка адреса первого элемента пары]
86 [30] A 40 [CONST1:] @ [увеличение его на единицу]
87 [31] T 2 [ADDR_START:] F [запись обновленного значения в ту же ячейку]
88 [32] A 4 [ADDR_END:] F [загрузка адреса второго элемента пары]
89 [33] S 40 [CONST1] @ [уменьшение его на единицу]
90 [34] T 4 [ADDR_END:] F [запись обновленного значения в ту же ячейку]
91 [35] E 2 [LOOP:] @ [возвращение в начало цикла]
92 [36] T 0 F [обнуление]
93 [37] [RETURN:] E 0 F [остановка программы]
94 [38] [TAKE:] A 0 F [инструкция для забора числа из нужной ячейки]
95 [39] [PUT:] T 0 F [инструкция для добавления числа в нужную ячейку]
96 [40] [CONST1:] P 0 D [константа, равная единице, для операций над адресами элементов]
97 [41] [c2:] P 1 F [константа, равная двойке для операций над длинной массива]
98 G K [директива IO2, фиксация начального адреса программы]
99 [0] A 11[LEN:] @ [загрузка в аккумулятор длины массива]
100 [1] T 0 F [запись длины массива во вспомогательную ячейку 1]

101 [2] A 9[ADDR_START:] @ [загрузка в аккумулятор адреса начала массива]
102 [3] T 2 F [запись адреса первого элемента во вспомогательную ячейку 2]
103 [4] A 10[ADDR_END:] @ [загрузка в аккумулятор адреса конца массива]
104 [5] T 4 F [запись адреса второго элемента во вспомогательную ячейку 3]
105 [6] A 6 @ [\ вызов]
106 [8] G 56 F [/ подпрограммы]
107 [8] Z 0 F
108 [9] [ADDR_START:] P 55 F [адрес начала массива]
109 [10] [ADDR_END:] P 57 F [адрес конца массива]
110 [11] [LEN:] P 2 F [длина массива]
111 [12] P 0 D [нулевой элемент массива, равный 1]
112 [13] P 1 F [первый элемент массива, равный 2]
113 [14] P 1 D [второй элемент массива, равный 3]
114 [15] P 2 D [третий элемент массива, равный 5]
115 EZ PF [директива IO2, переход к исполнению]

```

Результат работы

Исходный массив: 1(ячейка 111), 2(ячейка 112), 3(ячейка 113), 5(ячейка 114),
ожидаемый результат: 5(ячейка 111), 3(ячейка 112), 2(ячейка 113), 1(ячейка 114).



Результат совпал с ожидаемым.