Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**КУРСОВОЙ ПРОЕКТ**

**Дисциплина:** Низкоуровневое программирование

**Тема:** Машина Edsac

Выполнил

студент гр. 3530901/90003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Иванов Т.Р.

(подпись)

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

(подпись)

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

Санкт-Петербург   
2021

**ОГЛАВЛЕНИЕ**

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc58595498)

[МЕТОД РЕШЕНИЯ 3](#_Toc58595499)

[ПРОГРАММА INITIAL ORDERS 1 4](#_Toc58595499)

[РАБОТА ПРОГРАММЫ INITIAL ORDERS 1 6](#_Toc58595499)

[ПРОГРАММА INITIAL ORDERS 2 7](#_Toc58595499)

[РАБОТА ПРОГРАММЫ INITIAL ORDERS 2 10](#_Toc58595499)

[АДРЕСА И ПРАВИЛА КОДИРОВАНИЯ ИСХОДНЫХ ДАННЫХ И РЕЗУЛЬТАТОВ. РУКОВОДСТВО ПРОГРАММИСТА 12](#_Toc58595499)

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**Вариант 15**

В массиве поменять местами максимальный и минимальный элементы

**МЕТОД РЕШЕНИЯ**

Программа последовательно перебирает все элементы массива, сравнивая их с текущими максимальным и минимальным значениями. До работы с циклом как условный максимум и минимум принимается значение первого элемента массива. Дальше идет проверка, если текущий элемент массива меньше минимума, то присваиваем минимуму значение этого элемента и запоминаем его адрес. Аналогично с максимумом, если текущий элемент больше текущего максимума, то присваиваем максимуму значение этого элемента и запоминаем его адрес. После обработки всех элементов массива по адресу минимального элемента записываем значение максимального, а по адресу максимального – значение минимального.

Псевдокод:

int[] array;

int max, min, minAddr, maxAddr;

max = array[0];

min = array[0];

for (i = 0; i < array.size; i++){

if (array[i] <= min) {

min = array[i];

minAddr = i;

}

if (array[i] >= max) {

max = array[i];

maxAddr = i;

}

}

array[minAddr] = max;

array[maxAddr] = min;

}

# **ПРОГРАММА INITIAL ORDERS 1**

Нижу представлен текст программы для EDSAC, реализующей указанную функциональность и предполагающей загрузчик Initial Orders 1:

1. T141S
2. X0S [ для пошаговой отладки использовать Z0S ]
3. T0S [ запись аккумулятора в рабочую ячейку 0, обнуление аккумулятора ]
4. A125[<len>]S [ загрузка в аккумулятор длины обрабатываемого массива ]
5. T1S [ запись этого значения в ячейку 1 ]
6. A126[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]
7. L0L [ сдвиг аккумулятора на 1 разряд влево ]
8. A61[<r1>]S [ прибавляем код инструкции с нулевым полем адреса ]
9. T61[<r1>]S [ запись сформированной инструкции, обнуление аккумулятора ]
10. A126[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]
11. L0L [ сдвиг аккумулятора на 1 разряд влево ]
12. A127[<w1>]S [ прибавляем код инструкции с нулевым полем адреса ]
13. T127[<w1>]S [ запись сформированной инструкции, обнуление аккумулятора ]
14. A126[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]
15. L0L [ сдвиг аккумулятора на 1 разряд влево ]
16. A81[<r2>]S [ прибавляем код инструкции с полем адреса 1 ]
17. T81[<r2>]S [ запись сформированной инструкции, обнуление аккумулятора ]
18. A126[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]
19. L0L [ сдвиг аккумулятора на 1 разряд влево ]
20. A128[<w2>]S [ прибавляем код инструкции с полем адреса 1 ]
21. T128[<w2>]S [ запись сформированной инструкции, обнуление аккумулятора ]
22. A61[<r1>]S [ загрузка в аккумулятор кода инструкции ]
23. T54[<init>]S [ запись сформированной инструкции, обнуление аккумулятора ]
24. [init:]A0S [ загрузка в аккумулятор значения первого элемента массива ]
25. U129[<min>]S [ запись этого значения в ячейку значения минимального элемента, аккумулятор не обнуляется ]
26. T130[<max>]S [ запись этого значения в ячейку значения максимального элемента, обнуление аккумулятора ]
27. [loop:]A1S [ загружаем счетчик необработанных элементов массива ]
28. S124[<c2>]S [ уменьшаем на 2 ]
29. G118[<next>]S [ если результат меньше 0, переходим к следующему этапу ]
30. T1S [ обновляем значение счетчика и обнуляем аккумулятор ]
31. [r1:]A0S [ загрузка в аккумулятор значения из ячейки N ]
32. T0S [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]
33. A129[<min>]S [ загрузка в аккумулятор значения текущего минимального элемента ]
34. S0S [ вычитаем значение текущего элемента ]
35. G71[<g1>]S [ если результат меньше 0, переходим к следующей проверке ]
36. T2S [ обнуление аккумулятора ]
37. A0S [ загрузка в аккумулятор значения из ячейки N ]
38. T129[<min>]S [ запись его в ячейку значения текущего минимального элемента ]
39. A127[<w1>]S [ загрузка в аккумулятор кода инструкции ]
40. T122[<minaddr>]S [ записываем сформированную инструкцию в память ]
41. [g1:]T2S [ обнуление аккумулятора ]
42. A0S [ загрузка в аккумулятор значения из ячейки N ]
43. S130[<max>]S [ вычитаем значение текущего максимального элемента]
44. G80[<g2>]S [ если результат меньше 0, переходим к следующей проверке ]
45. T2S [ обнуление аккумулятора ]
46. A0S [ загрузка в аккумулятор значения из ячейки N ]
47. T130[<max>]S [ запись его в ячейку значения текущего максимального элемента ]
48. A127[<w1>]S [ загрузка в аккумулятор кода инструкции ]
49. T120[<maxaddr>]S [ записываем сформированную инструкцию в память ]
50. [g2:]T2S [ обнуление аккумулятора ]
51. [r2:]A1S [ загрузка в аккумулятор значения из ячейки N+1 ]
52. T0S [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]
53. A129[<min>]S [ загрузка в аккумулятор значения текущего минимального элемента ]
54. S0S [ вычитаем значение текущего элемента ]
55. G91[<g3>]S [ если результат меньше 0, переходим к следующей проверке ]
56. T2S [ обнуление аккумулятора ]
57. A0S [ загрузка в аккумулятор значения из ячейки N ]
58. T129[<min>]S [ запись его в ячейку значения текущего минимального элемента ]
59. A128[<w2>]S [ загрузка в аккумулятор кода инструкции ]
60. T122[<minaddr>]S [ записываем сформированную инструкцию в память ]
61. [g3:]T2S [ обнуление аккумулятора ]
62. A0S [ загрузка в аккумулятор значения из ячейки N ]
63. S130[<max>]S [ вычитаем значение текущего максимального элемента]
64. G100[<g4>]S [ если результат меньше 0, переходим к следующей проверке ]
65. T2S [ обнуление аккумулятора ]
66. A0S [ загрузка в аккумулятор значения из ячейки N ]
67. T130[<max>]S [ запись его в ячейку значения текущего максимального элемента ]
68. A128[<w2>]S [ загрузка в аккумулятор кода инструкции ]
69. T120[<maxaddr>]S [ записываем сформированную инструкцию в память ]
70. [g4:]T2S [ обнуление аккумулятора ]
71. A124[<c2>]S [ загрузка в аккумулятор константы 2 ]
72. L0L [ сдвиг на 1 разряд влево ]
73. A61[<r1>]S [ прибавляем код инструкции, исполненной на предыдущем шаге ]
74. T61[<r1>]S [ записываем сформированную инструкцию в память ]
75. A124[<c2>]S [ загрузка в аккумулятор константы 2 ]
76. L0L [ сдвиг на 1 разряд влево ]
77. A81[<r2>]S [ прибавляем код инструкции, исполненной на предыдущем шаге ]
78. T81[<r2>]S [ записываем сформированную инструкцию в память ]
79. A124[<c2>]S [ загрузка в аккумулятор константы 2 ]
80. L0L [ сдвиг на 1 разряд влево ]
81. A127[<w1>]S [ прибавляем код инструкции, исполненной на предыдущем шаге ]
82. T127[<w1>]S [ записываем сформированную инструкцию в память ]
83. A124[<c2>]S [ загрузка в аккумулятор константы 2 ]
84. L0L [ сдвиг на 1 разряд влево ]
85. A128[<w2>]S [ прибавляем код инструкции, исполненной на предыдущем шаге ]
86. T128[<w2>]S [ записываем сформированную инструкцию в память ]
87. E57[<loop>]S [ повторяем все операции; аккумулятор обнулен ]
88. [next:]T2S [ обнуление аккумулятора ]
89. A129[<min>]S [ загрузка в аккумулятор значения минимального элемента ]
90. [maxaddr:]T0S [ запись его в ячейку максимального элемента ]
91. A130[<max>]S [ загрузка в аккумулятор значения максимального элемента ]
92. [minaddr:]T0S [ запись его в ячейку минимального элемента ]
93. [exit:]Z0S [ останов ]
94. [c2]P1S [ 2 ]
95. [len:]P5S [ 10 ]
96. [addr:]P65L [ 131 ]
97. [w1:]T0S [ запись значения аккумулятора в ячейку с адресом N, обнуление аккумулятора ]
98. [w2:]T1S [ запись значения аккумулятора в ячейку с адресом N+1, обнуление аккумулятора ]
99. [min:]P0S [ значение минимального элемента ]
100. [max:]P0S [ значение максимального элемента ]
101. P0L [ 1 ]
102. P1S [ 2 ]
103. P1L [ 3 ]
104. P2S [ 4 ]
105. P2L [ 5 ]
106. P3S [ 6 ]
107. P3L [ 7 ]
108. P4S [ 8 ]
109. P4L [ 9 ]
110. P5S [ 10 ]

# **РАБОТА ПРОГРАММЫ INITIAL ORDERS 1**

Исходные данные:

• длина массива хранится в ячейке 125 и равна 10 элементам (рис. 1);

• адрес первой (нулевой) ячейки массива хранится в ячейке 126 и равен 131 (рис. 2);

• в данном случае массив расположен в ячейках 131–140. Значения в массиве: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] (рис. 3–12).



Рис. 1. Ячейка 125 — длина массива



Рис. 2. Ячейка 126 — адрес первого элемента массива



Рис. 3. Ячейка 131 — элемент массива [0]



Рис. 4. Ячейка 132 — элемент массива [1]



Рис. 5. Ячейка 133 — элемент массива [2]



Рис. 6. Ячейка 134 — элемент массива [3]



Рис. 7. Ячейка 135 — элемент массива [4]



Рис. 8. Ячейка 136 — элемент массива [5]



Рис. 9. Ячейка 137 — элемент массива [6]



Рис. 10. Ячейка 138 — элемент массива [7]



Рис. 11. Ячейка 139 — элемент массива [8]



Рис. 12. Ячейка 140 — элемент массива [9]

В результате работы программы максимальный и минимальный элементы массива меняют места. В данном примере поменяются первый и последний элементы массиве, то есть в ячейке 131 будет значение, которое было в ячейке 140, а в ячейке 140 – значение, что было в ячейке 131.



Рис. 13. Ячейка 131 — элемент массива [0]



Рис. 14. Ячейка 140 — элемент массива [9]

# **ПРОГРАММА INITIAL ORDERS 2**

Нижу представлен текст программы для EDSAC, реализующей указанную функциональность и предполагающей загрузчик Initial Orders 2:

T 56 K [ директива IO2, установка адреса загрузки ]

G K [ директива IO2, фиксация начального адреса подпрограммы ]

[ 0:] A 3 F [ пролог: формирование кода инструкции возврата в Acc ]

[ 1:] T 86 [<ret>] @ [ пролог: запись инструкции возврата ]

[ 2:] A 0 [<addr>] F [ загрузка в аккумулятор адреса 0-го элемента массива ]

[ 3:] A 23 [<r1>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 4:] T 23 [<r1>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[ 5:] A 0 [<addr>] F [ загрузка в аккумулятор адреса 0-го элемента массива ]

[ 6:] A 88 [<w1>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 7:] T 88 [<w1>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[ 8:] A 0 [<addr>] F [ загрузка в аккумулятор адреса 0-го элемента массива ]

[ 9:] A 43 [<r2>] @ [ прибавляем код инструкции с полем адреса 1 ]

[10:] T 43 [<r2>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[11:] A 0 [<addr>] F [ загрузка в аккумулятор адреса 0-го элемента массива ]

[12:] A 89 [<w2>] @ [ прибавляем код инструкции с полем адреса 1 ]

[13:] T 89 [<w2>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[14:] A 23 [<r1>] @ [ загрузка в аккумулятор кода инструкции ]

[15:] T 16 [<init>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[init:]

[16:] A 0 F [ загрузка в аккамулятор значения первого элемента массива ]

[17:] U 90 [<min>] @ [ запись этого значения в ячейку значения минимального элемента, аккумулятор не обнуляется ]

[18:] T 91 [<max>] @ [ запись этого значения в ячейку значения максимального элемента, обнуление аккумулятора ]

[loop:]

[19:] A 1 F [ загружаем счетчик необработанных элементов массива ]

[20:] S 87 [<c2>] @ [ уменьшаем на 2 ]

[21:] G 80 [<next>] @ [ если результат меньше 0, переходим к следующему этапу ]

[22:] T 1 F [ обновляем значение счетчика и обнуляем аккумулятор ]

[23:] [r1:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[24:] T 0 F [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]

[25:] A 90 [<min>] @ [ загрузка в аккумулятор значения текущего минимального элемента ]

[26:] S 0 F [ вычитаем значение текущего элемента ]

[27:] G 33 [<g1>] @ [ если результат меньше 0, переходим к следующей проверке ]

[28:] T 2 F [ обнуление аккумулятора ]

[29:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[30:] T 90 [<min>] @ [ запись его в ячейку значения текущего минимального элемента ]

[31:] A 88 [<w1>] @ [ загрузка в аккумулятор кода интсрукции ]

[32:] T 84 [<minaddr>] @ [ записываем сформированную инструкцию в память ]

[g1:]

[33:] T 2 F [ обнуление аккумулятора ]

[34:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[35:] S 91 [<max>] @ [ вычитаем значение текущего максимального элемента]

[36:] G 42 [<g2>] @ [ если результат меньше 0, переходим к следующей проверке ]

[37:] T 2 F [ обнуление аккумулятора ]

[38:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[39:] T 91 [<max>] @ [ запись его в ячейку значения текущего максимального элемента ]

[40:] A 88 [<w1>] @ [ загрузка в аккумулятор кода интсрукции ]

[41:] T 82 [<maxaddr] @ [ записываем сформированную инструкцию в память ]

[g2:]

[42:] T 2 F [ обнуление аккумулятора ]

[43:] [r2:] A 1 F [ загрузка в аккумулятор значения из ячейки N+1 ]

[44:] T 0 F [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]

[45:] A 90 [<min>] @ [ загрузка в аккумулятор значения текущего минимального элемента ]

[46:] S 0 F [ вычитаем значение текущего элемента ]

[47:] G 53 [<g3>] @ [ если результат меньше 0, переходим к следующей проверке ]

[48:] T 2 [<clean>] F [ обнуление аккумулятора ]

[49:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[50:] T 90 [<min>] @ [ запись его в ячейку значения текущего минимального элемента ]

[51:] A 89 [<w2>] @ [ загрузка в аккумулятор кода интсрукции ]

[52:] T 84 [<minaddr] @ [ записываем сформированную инструкцию в память ]

[g3:]

[53:] T 2 F [ обнуление аккумулятора ]

[54:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[55:] S 91 [<max>] @ [ вычитаем значение текущего максимального элемента]

[56:] G 62 [<g4>] @ [ если результат меньше 0, переходим к следующей проверке ]

[57:] T 2 F [ обнуление аккумулятора ]

[58:] A 0 F [ загрузка в аккумулятор значения из ячейки N ]

[59:] T 91 [<max>] @ [ запись его в ячейку значения текущего максимального элемента ]

[60:] A 89 [<w2>] @ [ загрузка в аккумулятор кода интсрукции ]

[61:] T 82 [<maxaddr] @ [ записываем сформированную инструкцию в память ]

[g4:]

[62:] T 2 F [ обнуление аккумулятора ]

[63:] A 87 [<c2>] @ [ загрузка в аккумулятор константы 2 ]

[64:] L 0 D [ сдвиг на 1 разряд влево ]

[65:] A 23 [<r1>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[66:] T 23 [<r1>] @ [ записываем сформированную инструкцию в память ]

[67:] A 87 [<c2>] @ [ загрузка в аккумулятор константы 2 ]

[68:] L 0 D [ сдвиг на 1 разряд влево ]

[69:] A 43 [<r2>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[70:] T 43 [<r2>] @ [ записываем сформированную инструкцию в память ]

[71:] A 87 [<c2>] @ [ загрузка в аккумулятор константы 2 ]

[72:] L 0 D [ сдвиг на 1 разряд влево ]

[73:] A 88 [<w1>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[74:] T 88 [<w1>] @ [ записываем сформированную инструкцию в память ]

[75:] A 87 [<c2>] @ [ загрузка в аккумулятор константы 2 ]

[76:] L 0 D [ сдвиг на 1 разряд влево ]

[77:] A 89 [<w2>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[78:] T 89 [<w2>] @ [ записываем сформированную инструкцию в память ]

[79:] E 19 [<loop>] @ [ повторяем все операции; аккумулятор обнулен ]

[next:]

[80:] T 2 F [ обнуление аккумулятора ]

[81:] A 90 [<min>] @ [ загрузка в аккамулятор значения минимального элемента ]

[82:] [maxaddr:] T 0 F [ запись его в ячейку максимального элемента ]

[83:] A 91 [<max>] @ [ загрузка в аккамулятор значения максимального элемента ]

[84:] [minaddr:] T 0 F [ запись его в ячейку минимального элемента ]

[exit:]

[85:] T 0 F [ обнуление аккумулятора ]

[ret:]

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

[87:] [c2:] P 1 F [ константа 2 ]

[88:] [w1:] T 0 F [ запись значения аккумулятора в ячейку с адресом N, обнуление аккумулятора ]

[89:] [w2:] T 1 F [ запись значения аккумулятора в ячейку с адресом N+1, обнуление аккумулятора ]

[90:] [min:] P 0 F [ значение минимального элемента ]

[91:] [max:] P 0 F [ значение максимального элемента ]

G K [ директива IO2, фиксация начального адреса программы ]

[ 0:] X 0 F [ для пошаговой отладки использовать Z 0 F ]

[ 1:] A 8 [<addr>] @ [ адрес массива (в начале работы обнуление аккумулятора не требуется) ]

[ 2:] T 0 F [ запись адреса массива в ячейку 0, обнуление аккумулятора ]

[ 3:] A 9 [<len>] @ [ длина массива ]

[ 4:] T 1 F [ запись длины массива в ячейку 1, обнуление аккумулятора ]

[ 5:] A 5 @ [\ вызов ]

[ 6:] G 56 [<sub>] F [/ подпрограммы ]

[ 7:] Z 0 F [ останов ]

[addr:]

[ 8:] P 10 [<array>] @ [ адрес массива = <Начало программы>+10 ]

[len:]

[ 9:] P 5 F [ длина массива - 10 ]

[array:]

[10:] P 0 D [ 1 ]

[11:] P 1 F [ 2 ]

[12:] P 1 D [ 3 ]

[13:] P 2 F [ 4 ]

[14:] P 2 D [ 5 ]

[15:] P 3 F [ 6 ]

[16:] P 3 D [ 7 ]

[17:] P 4 F [ 8 ]

[18:] P 4 D [ 9 ]

[19:] P 5 F [ 10 ]

EZ PF [ директива IO2, переход к исполнению ]

# **РАБОТА ПРОГРАММЫ INITIAL ORDERS 2**

Исходные данные:

* длина массива задана в строке 9 (от начала программы) и равна 10 элементам. В данном случае будет использована ячейка памяти под номером 157 (рис. 15);
* адрес первой (нулевой) ячейки массива задан в строке 8 (от начала программы) и равен 10 (от начала программы). В данном случае будет использована ячейка памяти под номером 156 (рис. 16);
* массив задан в строках 10–20 (от начала программы). Значения в массиве: [1, 2, 3, 4, 5, 6, 7, 8, 8, 10]. В данном случае будут использованы ячейки памяти под номерами 158–167 (рис. 17–26).



Рис. 15. Ячейка 157 — длина массива



Рис. 16. Ячейка 156 — адрес первого элемента массива



Рис. 17. Ячейка 158 — элемент массива [0]



Рис. 18. Ячейка 159 — элемент массива [1]



Рис. 19. Ячейка 160 — элемент массива [2]



Рис. 20. Ячейка 161 — элемент массива [3]



Рис. 21. Ячейка 162 — элемент массива [4]



Рис. 22. Ячейка 163 — элемент массива [5]



Рис. 23. Ячейка 164 — элемент массива [6]



Рис. 24. Ячейка 165 — элемент массива [7]



Рис. 25. Ячейка 166 — элемент массива [8]



Рис. 26. Ячейка 167 — элемент массива [9]

В результате работы программы максимальный и минимальный элементы массива меняют места. В данном примере поменяются первый и последний элементы массиве, то есть в ячейке 158 будет значение, которое было в ячейке 167, а в ячейке 167 – значение, что было в ячейке 158.



Рис. 27. Ячейка 158 — элемент массива [0]



Рис. 28. Ячейка 167 — элемент массива [9]

# 

# **АДРЕСА И ПРАВИЛА КОДИРОВАНИЯ ИСХОДНЫХ ДАННЫХ И РЕЗУЛЬТАТОВ. РУКОВОДСТВО ПРОГРАММИСТА**

1. *Программа для загрузчика Initial Orders 1.*

Чтобы воспользоваться программой, необходимо:

* указать длину массива в 125 строке;
* записать массив, начиная с 131 строки.

Результат выполнения программы будет записан в ячейки с адресами, соответствующими адресам элементов массива.

1. *Программа для загрузчика Initial Orders 2.*

Чтобы воспользоваться программой, необходимо:

* указать длину массива в 9 строке программы;
* записать массив, начиная с 10 строки программы.

Результат выполнения программы будет записан в ячейки со 158 строки. Это связано со смещением: запись подпрограммы в память EDSAC начинается с 56 ячейки (что указано в первой строке текста подпрограммы), подпрограмма занимает 92 строки, и плюс длина основной программа, после которой начинается массив (10 строк). Таким образом, в сумме получается, что первая строка массива находится в памяти по адресу 158, а не 10, как это записано в программе.