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 5](#_Toc58595499)

[ПРОГРАММА INITIAL ORDERS 2 7](#_Toc58595499)

[РАБОТА ПРОГРАММЫ INITIAL ORDERS 2 9](#_Toc58595499)

[АДРЕСА И ПРАВИЛА КОДИРОВАНИЯ ИСХОДНЫХ ДАННЫХ И РЕЗУЛЬТАТОВ. РУКОВОДСТВО ПРОГРАММИСТА 11](#_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. T104S
2. X0S [ для пошаговой отладки использовать Z0S ]
3. T0S [ запись аккумулятора в рабочую ячейку 0, обнуление аккумулятора ]
4. A89[<len>]S [ загрузка в аккумулятор длины обрабатываемого массива ]
5. T1S [ запись этого значения в ячейку 1 ]
6. A90[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]
7. L0L [ сдвиг аккумулятора на 1 разряд влево ]
8. A53[<r1>]S [ прибавляем код инструкции с нулевым полем адреса ]
9. T53[<r1>]S [ запись сформированной инструкции, обнуление аккумулятора ]
10. A90[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]
11. L0L [ сдвиг аккумулятора на 1 разряд влево ]
12. A91[<w1>]S [ прибавляем код инструкции с нулевым полем адреса ]
13. T91[<w1>]S [ запись сформированной инструкции, обнуление аккумулятора ]
14. A53[<r1>]S [ загрузка в аккумулятор кода инструкции ]
15. T46[<init>]S [ запись сформированной инструкции, обнуление аккумулятора ]
16. [init:]A0S [ загрузка в аккумулятор значения первого элемента массива ]
17. U92[<min>]S [ запись этого значения в ячейку значения минимального элемента, аккумулятор не обнуляется ]
18. T93[<max>]S [ запись этого значения в ячейку значения максимального элемента, обнуление аккумулятора ]
19. [loop:]A1S [ загружаем счетчик необработанных элементов массива ]
20. S88[<c1>]S [ уменьшаем на 1 ]
21. G82[<next>]S [ если результат меньше 0, переходим к следующему этапу ]
22. T1S [ обновляем значение счетчика и обнуляем аккумулятор ]
23. [r1:]A0S [ загрузка в аккумулятор значения из ячейки N ]
24. T0S [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]
25. A92[<min>]S [ загрузка в аккумулятор значения текущего минимального элемента ]
26. S0S [ вычитаем значение текущего элемента ]
27. G63[<g1>]S [ если результат меньше 0, переходим к следующей проверке ]
28. T2S [ обнуление аккумулятора ]
29. A0S [ загрузка в аккумулятор значения из ячейки N ]
30. T92[<min>]S [ запись его в ячейку значения текущего минимального элемента ]
31. A91[<w1>]S [ загрузка в аккумулятор кода инструкции ]
32. T86[<minaddr>]S [ записываем сформированную инструкцию в память ]
33. [g1:]T2S [ обнуление аккумулятора ]
34. A0S [ загрузка в аккумулятор значения из ячейки N ]
35. S93[<max>]S [ вычитаем значение текущего максимального элемента]
36. G72[<g2>]S [ если результат меньше 0, переходим к следующей проверке ]
37. T2S [ обнуление аккумулятора ]
38. A0S [ загрузка в аккумулятор значения из ячейки N ]
39. T93[<max>]S [ запись его в ячейку значения текущего максимального элемента ]
40. A91[<w1>]S [ загрузка в аккумулятор кода инструкции ]
41. T84[<maxaddr>]S [ записываем сформированную инструкцию в память ]
42. [g2:]T2S [ обнуление аккумулятора ]
43. A88[<c1>]S [ загрузка в аккумулятор константы 1 ]
44. L0L [ сдвиг на 1 разряд влево ]
45. A53[<r1>]S [ прибавляем код инструкции, исполненной на предыдущем шаге ]
46. T53[<r1>]S [ записываем сформированную инструкцию в память ]
47. A88[<c1>]S [ загрузка в аккумулятор константы 1 ]
48. L0L [ сдвиг на 1 разряд влево ]
49. A91[<w1>]S [ прибавляем код инструкции, исполненной на предыдущем шаге ]
50. T91[<w1>]S [ записываем сформированную инструкцию в память ]
51. E49[<loop>]S [ повторяем все операции; аккумулятор обнулен ]
52. [next:]T2S [ обнуление аккумулятора ]
53. A92[<min>]S [ загрузка в аккумулятор значения минимального элемента ]
54. [maxaddr:]T0S [ запись его в ячейку максимального элемента ]
55. A93[<max>]S [ загрузка в аккумулятор значения максимального элемента ]
56. [minaddr:]T0S [ запись его в ячейку минимального элемента ]
57. [exit:]Z0S [ останов ]
58. [c1]P0L [ 1 ]
59. [len:]P5S [ 10 ]
60. [addr:]P47S [ 94 ]
61. [w1:]T0S [ запись значения аккумулятора в ячейку с адресом N, обнуление аккумулятора ]
62. [min:]P0S [ значение минимального элемента ]
63. [max:]P0S [ значение максимального элемента ]
64. P0L [ 1 ]
65. P1S [ 2 ]
66. P1L [ 3 ]
67. P2S [ 4 ]
68. P2L [ 5 ]
69. P3S [ 6 ]
70. P3L [ 7 ]
71. P4S [ 8 ]
72. P4L [ 9 ]
73. P5S [ 10 ]

# **РАБОТА ПРОГРАММЫ INITIAL ORDERS 1**

Исходные данные:

• длина массива хранится в ячейке 89 и равна 10 элементам (рис. 1);

• адрес первой (нулевой) ячейки массива хранится в ячейке 80 и равен 94 (рис. 2);

• в данном случае массив расположен в ячейках 94–103. Значения в массиве: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] (рис. 3–12).



Рис. 1. Ячейка 89 — длина массива



Рис. 2. Ячейка 90 — адрес первого элемента массива



Рис. 3. Ячейка 94 — элемент массива [0]



Рис. 4. Ячейка 95 — элемент массива [1]



Рис. 5. Ячейка 96 — элемент массива [2]



Рис. 6. Ячейка 97 — элемент массива [3]



Рис. 7. Ячейка 98 — элемент массива [4]



Рис. 8. Ячейка 99 — элемент массива [5]



Рис. 9. Ячейка 100 — элемент массива [6]



Рис. 10. Ячейка 101 — элемент массива [7]



Рис. 11. Ячейка 102 — элемент массива [8]



Рис. 12. Ячейка 103 — 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 94 будет значение, которое было в ячейке 103, а в ячейке 103 – значение, что было в ячейке 94.



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



Рис. 14. Ячейка 103 — 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 52 [<ret>] @ [ пролог: запись инструкции возврата ]

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

[ 3:] A 17 [<r1>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 4:] T 17 [<r1>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

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

[ 6:] A 54 [<w1>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 7:] T 54 [<w1>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[ 8:] A 17 [<r1>] @ [ загрузка в аккумулятор кода инструкции ]

[ 9:] T 10 [<init>] @ [ запись сформированной инструкции, обнуление аккумулятора ]

[init:]

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

[11:] U 55 [<min>] @ [ запись этого значения в ячейку значения минимального элемента, аккумулятор не обнуляется ]

[12:] T 56 [<max>] @ [ запись этого значения в ячейку значения максимального элемента, обнуление аккумулятора ]

[loop:]

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

[14:] S 53 [<c1>] @ [ уменьшаем на 1 ]

[15:] G 46 [<next>] @ [ если результат меньше 0, переходим к следующему этапу ]

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

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

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

[19:] A 55 [<min>] @ [ загрузка в аккумулятор значения текущего минимального элемента ]

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

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

[22:] T 2 F [ обнуление аккумулятора ]

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

[24:] T 55 [<min>] @ [ запись его в ячейку значения текущего минимального элемента ]

[25:] A 54 [<w1>] @ [ загрузка в аккумулятор кода интсрукции ]

[26:] T 50 [<minaddr>] @ [ записываем сформированную инструкцию в память ]

[g1:]

[27:] T 2 F [ обнуление аккумулятора ]

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

[29:] S 56 [<max>] @ [ вычитаем значение текущего максимального элемента]

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

[31:] T 2 F [ обнуление аккумулятора ]

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

[33:] T 56 [<max>] @ [ запись его в ячейку значения текущего максимального элемента ]

[34:] A 54 [<w1>] @ [ загрузка в аккумулятор кода интсрукции ]

[35:] T 48 [<maxaddr] @ [ записываем сформированную инструкцию в память ]

[g2:]

[36:] T 2 F [ обнуление аккумулятора ]

[37:] A 53 [<c1>] @ [ загрузка в аккумулятор константы 1 ]

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

[39:] A 17 [<r1>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[40:] T 17 [<r1>] @ [ записываем сформированную инструкцию в память ]

[41:] A 53 [<c1>] @ [ загрузка в аккумулятор константы 1 ]

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

[43:] A 54 [<w1>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[44:] T 54 [<w1>] @ [ записываем сформированную инструкцию в память ]

[45:] E 13 [<loop>] @ [ повторяем все операции; аккумулятор обнулен ]

[next:]

[46:] T 2 F [ обнуление аккумулятора ]

[47:] A 55 [<min>] @ [ загрузка в аккамулятор значения минимального элемента ]

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

[49:] A 56 [<max>] @ [ загрузка в аккамулятор значения максимального элемента ]

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

[exit:]

[51:] T 0 F [ обнуление аккумулятора ]

[ret:]

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

[53:] [c1:] P 0 D [ константа 1 ]

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

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

[56:] [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 122 (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 121 (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 123–132 (рис. 17–26).



Рис. 15. Ячейка 122 — длина массива



Рис. 16. Ячейка 121 — адрес первого элемента массива



Рис. 17. Ячейка 123 — элемент массива [0]



Рис. 18. Ячейка 124 — элемент массива [1]



Рис. 19. Ячейка 125 — элемент массива [2]



Рис. 20. Ячейка 126 — элемент массива [3]



Рис. 21. Ячейка 127 — элемент массива [4]



Рис. 22. Ячейка 128 — элемент массива [5]



Рис. 23. Ячейка 129 — элемент массива [6]



Рис. 24. Ячейка 130 — элемент массива [7]



Рис. 25. Ячейка 131 — элемент массива [8]



Рис. 26. Ячейка 132 — 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 123 будет значение, которое было в ячейке 132, а в ячейке 132 – значение, что было в ячейке 123.



Рис. 27. Ячейка 123 — элемент массива [0]



Рис. 28. Ячейка 132 — 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 89 строке;
* записать массив, начиная с 94 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 123 строки. Это связано со смещением: запись подпрограммы в память EDSAC начинается с 56 ячейки (что указано в первой строке текста подпрограммы), подпрограмма занимает 57 строк, 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 123, а не 10, как это записано в программе.