

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

## **Отчёт по лабораторной работе № 2**

Дисциплина: Низкоуровневое программирование

Тема: Программирование EDSAC

Вариант 7

Выполнил студент гр. 3530901/90002 \_\_\_\_\_ Д.С. Ковалевский  
(подпись)

Принял старший преподаватель \_\_\_\_\_ Д.С. Степанов  
(подпись)

“ \_\_\_\_ ” \_\_\_\_\_ 2021 г.

Санкт-Петербург

2021

### **Цель работы:**

1. Разработать программу для EDSAC, реализующую определенную вариант 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 вариант 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.

## Вариант 7: Определение k-й порядковой статистики in-place.

### 1. Постановка задачи

Необходимо смоделировать программу для EDSAC, которая определит такой элемент неупорядоченного массива, если бы он был k-ым в упорядоченном.

Для реализации будем использовать сортировку и вывод k-ого (k от 0 до n) элемента уже отсортированного массива.

### 2. Initial Orders 1.

Программа для EDSAC, реализующий поиск k-й порядковой статистики, использующий загрузчик Initial Orders 1.

```
1      [обнуление iter, bool]
2      [31] T 120[end] S
3      [32] [w4] T 0 S
4      [33] T 117[iter] S
5      [34] T 118[bool] S
6
7      [ если второе число меньше первого]
8      [35] [w2] A 110[N2] S [c1]
9      [36] S 109[N1] S [c2]
10     [37] E 51[w1] S
11
12     [ проверка не является ли второе число 0]
13     [38] T 0 S
14     [39] A 110[N2] S [c7]
15     [40] S 115[1] S
16     [41] G 77[w3] S
17
18     [меняем местами i и i+1]
19     [42] T 0 S
20     [43] A 115[1] S
21     [44] T 118[bool] S
22     [45] A 109[N1] S [c3]
23     [46] T 116[temp] S
24     [47] A 110[N2] S [c4]
25     [48] T 109[N1] S [c5]
26     [49] A 116[temp] S
27     [50] T 110[N2] S [c6]
28
29     [+1 к адресам для продвижения по итерации]
30     [51] [w1] T 0 S
31     [52] A 117[iter] S
32     [53] A 115[1] S
33     [54] T 117[iter] S
34     [55] A 35[c1] S
35     [56] A 115[1] S
36     [57] T 35[c1] S
37     [58] A 36[c2] S
38     [59] A 115[1] S
39     [60] T 36[c2] S
40     [61] A 45[c3] S
41     [62] A 115[1] S
42     [63] T 45[c3] S
43     [64] A 47[c4] S
44     [65] A 115[1] S
45     [66] T 47[c4] S
46     [67] A 48[c5] S
```

Рис.1. Программа на Ю 1 часть 1.

```

47 [68] A 115[l] S
48 [69] T 48[c5] S
49 [70] A 50[c6] S
50 [71] A 115[l] S
51 [72] T 50[c6] S
52 [73] A 39[c7] S
53 [74] A 115[l] S
54 [75] T 39[c7] S
55 [76] E 35[w2] S
56
57 [обнуление для следующей итерации]
58 [77] [w3] T 0 S
59 [78] A 35[c1] S
60 [79] S 117[iter] S
61 [80] T 35[c1] S
62 [81] A 36[c2] S
63 [82] S 117[iter] S
64 [83] T 36[c2] S
65 [84] A 45[c3] S
66 [85] S 117[iter] S
67 [86] T 45[c3] S
68 [87] A 47[c4] S
69 [88] S 117[iter] S
70 [89] T 47[c4] S
71 [90] A 48[c5] S
72 [91] S 117[iter] S
73 [92] T 48[c5] S
74 [93] A 50[c6] S
75 [94] S 117[iter] S
76 [95] T 50[c6] S
77 [96] A 39[c7] S
78 [97] S 117[iter] S
79 [98] T 39[c7] S
80 [99] A 118[bool] S
81 [100] S 115[l] S
82 [101] G 103[w5] S
83 [102] E 32[w4] S
84
85 [вывод нужного числа в 120 ячейку]
86 [103] [w5] T 0 S
87 [104] A 107[toExit] S
88 [105] A 119[k] S
89 [106] T 107[toExit] S
90 [107] A 109[Nl] S [toExit]
91 [108] T 120[k] S
92
93 [константы]
94 [109] P 100 S [N1]
95 [110] P 70 S [N2]
96 [111] P 40 S
97 [112] P 10 S
98 [113] P 20 S
99 [114] P 0 S
100 [115] P 1 S [l]
101 [116] P 0 S [temp]
102 [117] P 1 S [iter]
103 [118] P 0 S [bool]
104 [119] P 0 S [k]

```

Рис.2. Программа на Ю 1 часть 2.

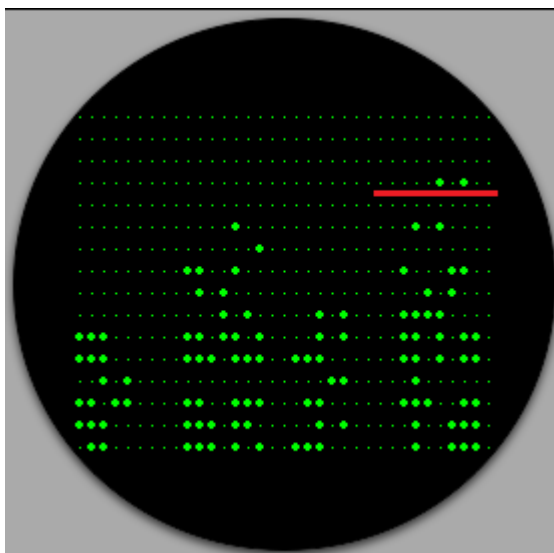


Рис.3. Результат работы программы на Ю 1.

WORD 120 Order = P 10 S Integer 120S = 20 Fraction 120S = 0.000305

Рис.4. Значение в ячейке 120 в конце работы программы.

На рис.3 и 4 видно, что в ячейке с выводом оказалось число P10S.

В массиве изначально находятся P100S, P70S, P40S, P10S, P20S. В порядке сортировки P10S имеет индекс  $k = 0$ , который мы и искали, указав его в 119 ячейке.

Входной массив данных храниться в ячейках 109 – 113, а также в ячейке 114 хранится 0, означающий конец массива, таким образом в массив нельзя добавлять нули, массив должен состоять из положительных десятичных ненулевых чисел.

Число  $k$  вводится в ячейку 119, в конце работы в ней же будет выведен результат, таким образом для повторного запуска программы необходим сброс EDSAC.

Для увеличения массива, необходимо сместить все строки программы, вниз на необходимое число, а также поменять значения адресов констант в ячейках, где они используются (номер строки + 30).

### 3.Initial Orders 2.

Та же программа, используется как замкнутая подпрограмма с тестовой программой, вызывающей её.

```
1      [обнуление iter, bool]
2  [41] T 43[start] K
3  [42] G K [фиксируем адрес]
4  [43] [0] A 3 F [инструкция возврата]
5  [44] [1] T [выход] @
6  [45] [2] [w4] T 0 F
7  [46] [3] T 88[iter] @
8  [46] [4] T 89[bool] @
9
10     [ если второе число меньше первого]
11  [47] [5] [w2] A 81[N2] @ [c1]
12  [48] [6] S 80[N1] @ [c2]
13  [49] [7] E 21[w1] @
14
15     [ проверка не является ли второе число 0]
16  [50] [8] T 0 F
17  [51] [9] A 81[N2] @ [c7]
18  [52] [10] S 86[1] @
19  [53] [11] G 47[w3] @
20
21     [меняем местами i и i+1]
22  [54] [12] T 0 F
23  [55] [13] A 86[1] @
24  [56] [14] T 89[bool] @
25  [57] [15] A 80[N1] @ [c3]
26  [58] [16] T 87[temp] @
27  [59] [17] A 81[N2] @ [c4]
28  [60] [18] T 80[N1] @ [c5]
29  [61] [19] A 87[temp] @
30  [62] [20] T 81[N2] @ [c6]
31
32     [+1 к адресам для продвижения по итерации]
33  [63] [21] [w1] T 0 F
34  [64] [22] A 88[iter] @
35  [65] [23] A 86[1] @
36  [66] [24] T 88[iter] @
37  [67] [25] A 5[c1] @
38  [68] [26] A 86[1] @
39  [69] [27] T 5[c1] @
40  [70] [28] A 6[c2] @
41  [71] [29] A 86[1] @
42  [72] [30] T 6[c2] @
43  [73] [31] A 15[c3] @
44  [74] [32] A 86[1] @
45  [75] [33] T 15[c3] @
46  [76] [34] A 17[c4] @
47  [77] [35] A 86[1] @
48  [78] [36] T 17[c4] @
49  [79] [37] A 18[c5] @
50  [80] [38] A 86[1] @
51  [81] [39] T 18[c5] @
52  [82] [40] A 20[c6] @
53  [83] [41] A 86[1] @
54  [84] [42] T 20[c6] @
55  [85] [43] A 9[c7] @
56  [86] [44] A 86[1] @
57  [87] [45] T 9[c7] @
```

Рис.5. Программа на Ю 2 часть 1.

```

58 [88] [46] E 5[w2] @
59
60 [обнуление для следующей итерации]
61 [89] [47] [w3] T 0 F
62 [90] [48] A 5[c1] @
63 [91] [49] S 88[iter] @
64 [92] [50] T 5[c1] @
65 [93] [51] A 6[c2] @
66 [94] [52] S 88[iter] @
67 [95] [53] T 6[c2] @
68 [96] [54] A 15[c3] @
69 [97] [55] S 88[iter] @
70 [98] [56] T 15[c3] @
71 [99] [57] A 17[c4] @
72 [100] [58] S 88[iter] @
73 [101] [59] T 17[c4] @
74 [102] [60] A 18[c5] @
75 [103] [61] S 88[iter] @
76 [104] [62] T 18[c5] @
77 [105] [63] A 20[c6] @
78 [106] [64] S 88[iter] @
79 [107] [65] T 20[c6] @
80 [108] [66] A 9[c7] @
81 [109] [67] S 88[iter] @
82 [110] [68] T 9[c7] @
83 [111] [69] A 89[bool]@
84 [112] [70] S 86[l] @
85 [113] [71] G 73[w5] @
86 [114] [72] E 2[w4] @
87
88 [вывод нужного числа в 120 ячейку]
89 [115] [73] [w5] T 0 F
90 [116] [74] A 77[toExit] @
91 [117] [75] A 90[k] @
92 [118] [76] T 77[toExit] @
93 [119] [77] A 80[N1] @ [toExit]
94 [120] [78] T 138 F
95 [121] [79] E 0 F [возврат из подпрограммы]
96
97 [константы]
98 [122] [80] P 100 F [N1]
99 [123] [81] P 70 F [N2]
100 [124] [82] P 40 F
101 [125] [83] P 10 F
102 [126] [84] P 20 F
103 [127] [85] P 0 F
104 [128] [86] P 1 F [l]
105 [129] [87] P 0 F [temp]
106 [130] [88] P 1 F [iter]
107 [131] [89] P 0 F [bool]
108 [132] [90] P 3 F [k]
109 [133] [91] G K [фиксируем адрес]
110 [134] [1] A 0 @
111 [135] [2] G 43 F
112 [136] [3] EZ PF [завершение]

```

Рис.6. Программа на Ю 2 часть 2.

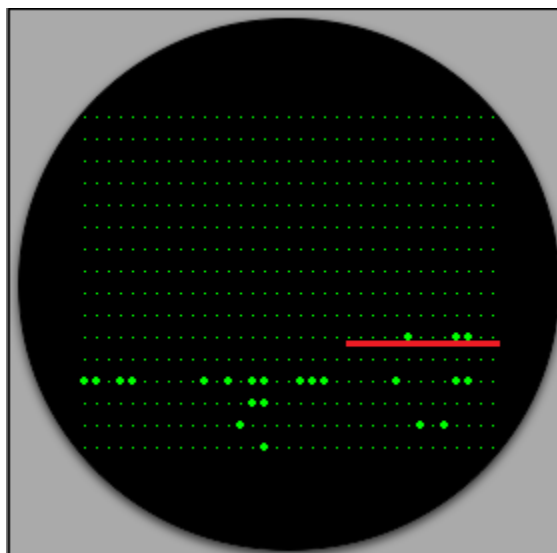


Рис.7. Результат работы программы на Ю 2.

WORD 138 Order = P 70 F Integer 138F = 140 Fraction 138F = 0.002136

Рис.8. Значение в ячейке 138 в конце работы программы.

На рис.7 и 8 видно, что в ячейке с выводом оказалось число P70F. В массиве изначально находятся P100F, P70F, P40F, P10F, P20F. В порядке сортировки P70F имеет индекс  $k = 3$ , который мы и искали, указав его в 132 ячейке.

Входной массив данных храниться в ячейках 122 – 126, а также в ячейке 127 хранится 0, означающий конец массива, таким образом в массив нельзя добавлять нули, массив должен состоять из положительных десятичных ненулевых чисел.

Число  $k$  вводится в ячейку 132, в конце работы в ней же будет выведен результат, таким образом для повторного запуска программы необходим сброс EDSAC.

Для увеличения массива, необходимо сместить все строки программы, вниз на необходимое число, а также поменять значения адресов констант в ячейках, где они используются (с учетом использование теты, число строки – 3 для модификации программы).

Результат выводится в 138 ячейку для удобства нахождения.



#### **4. Алгоритм работы определения k-ой статистики**

- 1 шаг. Обнуление констант booling и iteration
- 2 шаг. Вычитание из  $i + 1$  числа  $i$ .
- 3 шаг. Если число положительное, то шаг 6.
- 4 шаг. Проверяем, не является ли второе число нулем, если да, то шаг 8.
- 5 шаг. Устанавливаем booling в 1, меняем местами числа  $i$  и  $i+1$ .
- 6 шаг. В значения ячеек, где используются  $i$  и  $i+1$ , а так же к iteration прибавляется единица.
- 7 шаг. Переход на шаг 2.
- 8 шаг. Вычитание из ячеек, где используются  $i$  и  $i+1$  числа iteration для следующего прохода по массиву.
- 9 шаг. Если число booling равно единице, то переход на шаг 1.
- 10 шаг. Прибавление к индексу ячейки N1 числа k
- 11 шаг. Вывод нужного числа.

#### **5. Вывод**

Составленные программы для EDSAC с использованием IO 1 и IO 2 удовлетворяют варианту и находят k-ую порядковую статистику in-place.