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 вариантом 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 вариантом 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.

```
[31] T 116[end] S
[32] [w4]T 113[booling] S [Сброс bool в 0]
[33] T 114[iter] S [ Установка в 0 счетчика итераций]
[34] [w1]A 106[N1] S [i - ое число массива]
[35] S 107[N2] S [i + 1 - число массива]
[36] G 50[to w2] S [if i < i+1]
[37] S 106[N1] S [i2 - проверка, ноль ли во втором числе ]
[38] S 110[1] S [null]
[39] E 76[to w3] S [если второе число ноль - переход на следующий круг]
[40] T 1 S [clear acc]
[41] A 111[1]S       [\]
[42] T 113[booling]S [/ установка bool в 1]
[43] T 1 S [clear acc]
[44] A 106[N1] S [copy i]
[45] T 112[temp]S [paste to temp]
[46] A 107[N2] S [copy i+1]
[47] T 106[N1] S [paste to N1]
[48] A 112[temp]S [copy temp]
[49] T 107[N2] S [Paste to N2]
[50] [w2]T 1 S [clear acc]
[51] A 34[i] S [\]
[52] A 111[1] S[| (i) += 1]
[53] T 34[i] S [/]
[54] A 35[i + 1] S [\]
[55] A 111[1] S    [| (i+1) +=1]
[56] T 35[i + 1] S [/]
[57]  A 44[copy i] S [\]
[58] A 111[1] S     [| (copy i) +=1]
[59] T 44[copy i] S [/]
[60] A 46[copy i + 1] S [\]
[61] A 111[1] S         [| (copy i + 1) +=1]
[62] T 46[copy i + 1] S [/]
[63] A 47[paste to N1] S [\]
[64] A 111[1] S          [| (paste to N1) +=1]
[65] T 47[paste to N1] S [/]
[66] A 49[paste to N2] S [\]
[67] A 111[1] S          [| (paste to N2) +=1]
[68] T 49[paste to N2] S [/]
[69] A 37[i2] S [\]
[70] A 111[1] S [| (i2) +=1]
[71] T 37[i2] S [/]
[72] A 114[iteration] S [\]
[73] A 111[1] S         [| iter += 1]
```

Рис.1. Программа на IO 1 часть 2.

```
[74] T 114[iteration] S [/]
[75] E 34[to w1] S [переход к следующей итерации]
[76] [w3]A 34[i] S          [\]
[77] S 114[iter] S          [|]
[78] T 34[i] S              [|]
[79] A 35[i + 1] S          [|]
[80] S 114[iter] S          [|]
[81] T 35[i + 1] S          [|]
[82] A 37[i2] S             [|]
[83] S 114[iter] S          [|]
[84] T 37[i2] S             [|]
[85] A 44[copy i] S         [| сброс всех индексов для слудующего круга]
[86] S 114[iter] S          [|]
[87] T 44[copy i] S         [|]
[88] A 46[copy i+1] S       [|]
[89] S 114[iter] S          [|]
[90] T 46[copy i+1] S       [|]
[91] A 47[paste to N1] S    [|]
[92] S 114[iter] S          [|]
[93] T 47[paste to N1] S    [|]
[94] A 49[paste to N2] S    [|]
[95] S 114[iter] S          [|]
[96] T 49[paste to N2] S    [/]
[97] A 113[bool] S      [\]
[98] S 111[1] S         [| если  перестановок не было, то переходим к выводу]
[99] E 32[w4] S         [/]
[100] T 1 S [clear acc]
[101] A 115[k] S          [\]
[102] A 104[exitpath] S   [| считаем индекс k-ого число]
[103] T 104[exitpath] S   [/]
[104] A 106[N1] S [exitpath]
[105] T 115[k] S    [выводим в 115 ячейку результат]
[106] [N1]P 100 S       [\]
[107] [N2]P 10 S        [|]
[108] [N3]P 35 S        [| массив данных]
[109] [N4]P 22 S        [/]
[110] [null]P 0 S       []
[111] [1]P 1 S          [\]
[112] [temp]P 0 S       [| константы]
[113] [booling]P 0 S    [| показывает были ли изменения в итерации]
[114] [iteration]P 0 S[/]
[115] [k]P 2 S          [ число k, и результат после работы программы]
```

Рис.2. Программа на IO 1 часть 2.

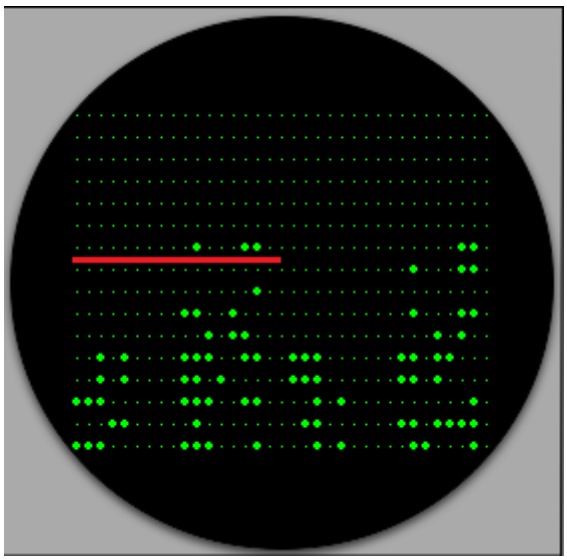

Рис.3. Результат работы программы на IO 1.

WORD 115 Order = P 35 S Integer 115S = 70 Fraction 114L = 0.00106811558

Рис.4. Значение в ячейке 115 в конце работы программы.

На рис.3 и 4 видно, что в ячейке с выводом оказалось число P35S. В массиве изначально находятся P100S, P10S, P35S, P22S. В порядке сортировки P35S имеет индекс k = 2, который мы и искали, указав его в 115 ячейке.

Входной массив данных храниться в ячейках 106 – 109(строки 76-79), а также в ячейке 110(строка 80) хранится 0, означающий конец массива, таким образов в массив нельзя добавлять нули, массив должен состоять из положительных десятичных ненулевых чисел.

Число k вводится в ячейку 115(85 строка), 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, начиная с 80 вниз на необходимое число, а также поменять значения адресов констант в ячейках, где они используются (номер строки + 30).

Программа использует сортировку пузырьком, со строки 2 по 69. В строках 14 -19 ячейки меняются местами, если необходимо, все остальные строки нужны для смены в них индексов или сброса к первоначальному.

67-75 выводит результат, прибавляя к индексу первого числа массива, число k.

3.Initial Orders 2.

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

```
[41] T 43[start] K
[42] G  K [фиксируем адрес]
[43] [0] A 3 F [инструкция возврата]
[44] [1] T 76 @ [запись инструкции возврата]
[45] [2] [w4]T 84[booling] @ [Сброс bool в 0]
[46] [3] T 85[iter] @ [ Установка в 0 счетчика итераций]
[47] [4] [w1]A 77[N1] @ [i - ое число массива]
[48] [5]S 78[N2] @ [i + 1 - число массива]
[49] [6]G 20[to w2] @ [if i < i+1]
[50] [7] S 77[N1] @ [i2 - проверка, ноль ли во втором числе ]
[51] [8] S 81[1] @ [null]
[52] [9] E 46[to next loop w3] @ [если второе число ноль - переход на следующий круг ]
[53] [10] T 1 F [cl acc]
[54] [11] A 82[1]@       [\]
[55] [12] T 84[booling]@ [/ установка bool в 1]
[56] [13] T 1 F [clear acc]
[57] [14] A 77[N1] @ [copy i]
[58] [15] T 83[temp]@ [paste to temp]
[59] [16] A 78[N2] @ [copy i +1]
[60] [17] T 77[N1] @ [paste to N1]
[61] [18] A 83[temp]@ [copy temp]
[62] [19] T 78[N2] @ [Paste to N2]
[63] [20] [w2]T 1 F [clear acc]
[64] [21] A 4[i] @ [\]
[65] [22] A 82[1] @   [| i += 1]
[66] [23] T 4[i] @ [/]
[67] [24] A 5[i + 1] @ [\]
[68] [25] A 82[1] @       [| i+1 +=1]
[69] [26] T 5[i + 1] @ [/]
[70] [27] A 14[copy i] @ [\]
[71] [28] A 82[1] @        [| copy i +=1]
[72] [29] T 14[copy i] @ [/]
[73] [30] A 16[copy i + 1] @ [\]
[74] [31] A 82[1] @            [| copy i + 1 +=1]
[75] [32] T 16[copy i + 1] @ [/]
[76] [33] A 17[paste to N1] @ [\]
[77] [34] A 82[1] @             [| paste to N1+=1]
[78] [35] T 17[paste to N1] @ [/]
[79] [36] A 19[paste to N2] @ [\]
[80] [37] A 82[1] @             [| paste to N2+=1]
[81] [38] T 19[paste to N2] @ [/]
[82] [39] A 7[i2] @ [\]
```

Рис.5. Программа на IO 2 часть 1.

```
[83] [40] A 82[1] @    [| i2 +=1]
[84] [41] T 7[i2] @ [/]
[85] [42] A 85[iteration] @ [\]
[86] [43] A 82[1] @         [| iter += 1]
[87] [44] T 85[iter] @      [/]
[88] [45] E 4[to w1] @      [переход к следующей итерации]
[89] [46] [w3]A 4[i] @       [\]
[90] [47] S 85[iter] @       [|]
[91] [48] T 4[i] @           [|]
[92] [49] A 5[i + 1] @       [|]
[93] [50] S 85[iter] @       [|]
[94] [51] T 5[i + 1] @       [|]
[95] [52] A 7[i2] @          [|]
[96] [53] S 85[iter] @       [|]
[97] [54] T 7[i2] @          [| сброс всех индексов для слудующего круга]
[98] [55] A 14[copy i] @     [|]
[99] [56] S 85[iter] @       [|]
[100] [57] T 14[copy i] @     [|]
[101] [58] A 16[copy i+1] @   [|]
[102] [59] S 85[iter] @       [|]
[103] [60] T 16[copy i+1] @   [|]
[104] [61] A 17[paste to N1] @[|]
[105] [62] S 85[iter] @       [|]
[106] [63] T 17[paste to N1] @ [|]
[107] [64] A 19[paste to N2] @ [|]
[108] [65] S 85[iter] @        [|]
[109] [66] T 19[paste to N2] @ [/]
[110] [67] A 84[bool] @      [\]
[111] [68] S 82[1] @         [| если  перестановок не было, то переходим к выводу]
[112] [69] E 2[w4] @         [/]
[113] [70] [w5]T 1 F [clear acc]
[114] [71] A 86[k] @          [\]
[115] [72] A 74[exitpath] @   [| считаем индекс k-ого числа]
[116] [73] T 74[exitpath] @   [/]
[117] [74] A 77[N1] @ [exitpath]
[118] [75] T 134[k] F         [выводим в 134 ячейку результат]
[119] [76] E 0 F [возврат из подпрограммф]
[120] [77] [N1]P 100 F    [\]
[121] [78] [N2]P 43 F     [|]
[122] [79] [N3]P 35 F     [| массив данных]
[123] [80] [N4]P 50 F     [/]
[124] [81] [null]P 0 F
```

Рис.6. Программа на IO 2 часть 2.

```
85  [125] [82] [l]P 1 F         [\]
86  [126] [83] [temp]P 0 F      [| константы]
87  [127] [84] [booling]P 0 F   [| показывает были ли изменения в итерации]
88  [128] [85] [iteration]P 0 F [/]
89  [129] [86] [k]P 2 F         [ число k]
90  [130] [87] G   K [фиксируем адрес]
91  [131] [1] A 0 @  [\вызов подпрограммы]
92  [132] [2] G 43 F [/]
93  [133] [3] Z 0 F  [останов]
94  [134] [4] EZ PF  [завершение]
```

Рис.7. Программа на IO 2 часть 3.

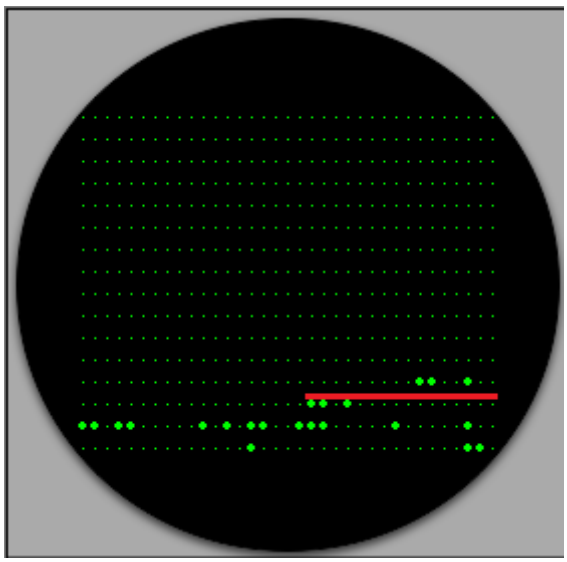

Рис.8. Результат работы программы на IO 2.

```
WORD 134   Order = P 50 F   Integer 134F = 100   Fraction 134F = 0.001526
```

Рис.9. Значение в ячейке 134 в конце работы программы.

На рис.8 и 9 видно, что в ячейке с выводом оказалось число P50F. В массиве изначально находятся P100F, P43F, P35F, P50F. В порядке сортировки P50F имеет индекс k = 2, который мы и искали, указав его в 129 ячейке.

Входной массив данных храниться в ячейках 120 – 123(строки 80-83), а также в ячейке 128(строка 88) хранится 0, означающий конец массива, таким образов в массив нельзя добавлять нули, массив должен состоять из положительных десятичных ненулевых чисел.

Число k вводится в ячейку 119(89 строка), 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, начиная с 84 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 134 ячейку для удобства нахождения.

4. Алгоритм работы определения k-ой статистики

1 шаг. Обнуление констант booling и iteration

2 шаг. Вычитание из i числа i+1.

3 шаг. Если число отрицательное, то шаг 6.

4 шаг. Вычитаем еще раз первое число, если в аккумуляторе положительное число, 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.