

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе №2

по дисциплине «Низкоуровневое программирование»

Машина EDSAC

Работу

выполнил:

Ильин В.П.

Группа:

35300901/10005

Преподаватель:

Коренев Д.А.

Санкт-Петербург
2022

Содержание

1. Техническое задание	3
2. Метод решения	3
3. Программа Orders1	3
3.1. Код программы Orders1	5
3.2. Работа программы Orders1	7
4. Программа Orders2	7
4.1. Код программы Orders2	8
4.2. Работа программы Orders2	10

1. Техническое задание

Написать программу поиска k -й порядковой статистики массива in-place для машины EDSAC.

2. Метод решения

По определению, k -й порядковой статистикой данного массива называется k -й элемент этого массива, если бы он был отсортирован. Будем искать ее по определению: отсортируем массив синтаксически простейшей сортировкой и возьмем элемент с индексом $(k - 1)$.

3. Программа Orders1

Для сортировки в программе используются два цикла. Внутренний цикл проходит по массиву и сравнивает соседние элементы: если правый элемент оказывается меньше первого, то они меняются местами; внешний же цикл только обеспечивает достаточное количество повторений внутреннего. Синтаксически, в машине EDSAC работа циклов не реализована, поэтому создадим для обоих циклов счетчики количества итераций и поместим их в ячейки 1 и 2. Изначально, эти счетчики будут равны (длине массива -2). Это число содержится в ячейке с меткой $\langle len \rangle$ под номером 124.

Внутри фрагмента реализации перестановки двух соседних элементов массива используются инструкции чтения и записи по адресам массива. На каждой итерации эти адреса нужно будет менять, поэтому помимо прочего, перед сортировкой необходимо инициализировать эти инструкции и куда-то их сохранить. Для этого, к каждой инструкции будем добавлять адрес массива, содержащийся в ячейке с меткой $\langle addr \rangle$ под номером 123, предварительно сдвинув его на 1 разряд влево. Сформированные инструкции будем записывать в ячейки 10–13, а также обновлять их под соответствующими метками: $\langle r_1 \rangle$, $\langle r_2 \rangle$, $\langle w_1 \rangle$, $\langle w_2 \rangle$.

Далее идем сам алгоритм сортировки. Во внешнем цикле сначала проверим условие выхода из него: считаем счетчик количества оставшихся итераций и, если он меньше нуля, то выйдем из цикла, перейдя к метке $\langle outer_for_exit \rangle$. В противном же случае программа продолжит работу и войдет во внутренний цикл. В нем она так же сначала проверит условие выхода и, если оно не выполнится, то перейдет к чтению текущих интересующих нас элементов массива, которые будут записаны в ячейки 5 и 6. При записи правого элемента, для уменьшения количества действий аккумулятор не обнуляется. Для проверки на необходимость переставить элементы рассматривается разность $a_{j+1} - a_j \geq 0$. Если это условие выполняется, то осуществляется переход к метке $\langle skip_swap \rangle$, иначе осуществляется перестановка.

В конце каждой итерации внутреннего цикла обновляются инструкции чтения и записи – к каждой прибавляется сдвинутая на один разряд влево единица, а также единица вычитается из счетчика оставшихся итераций.

Если же во внутреннем цикле сработал переход к метке $\langle inner_for_exit \rangle$, то значит на этом шаге внешнего цикла все необходимые итерации внутреннего уже были выполнены, и надо переходить к следующей. Для этого, во-первых, восстанавливается счетчик оставшихся итераций внутреннего цикла, во-вторых, единица вычитается из счетчика для внешнего цикла, и, в-третьих, при помощи сохраненных заранее в ячейки 10–13 копий, восстанавливаются инструкции чтения и записи. Осуществляется переход в начало внешнего цикла.

После перехода к метке $\langle outer_for_exit \rangle$, циклы закончатся, а массив будет отсортирован. Все, что остается – найти a_{k-1} . Для этого в аккумулятор добавляется адрес массива,

после чего к нему прибавляется k и вычитается единица. После этого, содержимое аккумулятора сдвигается на один разряд влево и прибавляется к инструкции чтения ответа: $\langle r_{ans} \rangle$.

Ответ записывается в ячейку $\langle res \rangle$ под номером 126, и программа завершает работу.

3.1. Код программы Orders1

```
[ Вариант №5 ]
[ Поиск k-й порядковой статистики ]

[ Ячейка для k: (k)125 ]
[ Ячейка для ответа: (res)126 ]
[ Ячейка для длины массива: (len)124 ]
[ Адрес исходного (в последствии - отсортированного) массива: 127 ]

[31] T [end:]135 S
[32] X 0 S

[ записать начальные значения счетчиков для циклов ]
[33] A [len]124 S
[34] T 1 S [ количество оставшихся итераций внешнего цикла ]
[35] A [len]124 S
[36] T 2 S [ количество оставшихся итераций внутреннего цикла ]

[ добавить адрес массива для инструкций чтения и записи элементов ]
[37] A [addr]123 S
[38] L 0 L
[39] A [r1]65 S
[40] U 10 S [ сохранить инструкцию, чтобы потом можно было к ней вернуться ]
[41] T [r1]65 S
[42] A [addr]123 S
[43] L 0 L
[44] A [r2]67 S
[45] U 11 S [ сохранить инструкцию, чтобы потом можно было к ней вернуться ]
[46] T [r2]67 S
[47] A [addr]123 S
[48] L 0 L
[49] A [w1]73 S
[50] U 12 S [ сохранить инструкцию, чтобы потом можно было к ней вернуться ]
[51] T [w1]73 S
[52] A [addr]123 S
[53] L 0 L
[54] A [w2]75 S
[55] U 13 S [ сохранить инструкцию, чтобы потом можно было к ней вернуться ]
[56] T [w2]75 S

[ сортировка ]

[outer_for:]
[57] A 1 S
[58] G [outer_for_exit]112 S [ количество оставшихся итераций < 0 ? ]

[inner_for:]
[59] A 2 S
[60] G [inner_for_exit:]97 S [ количество оставшихся итераций < 0 ? ]
[61] T 0 S
[62] A [len]124 S
[63] S 2 S
[64] T 4 S [ j ]
[65] [r1:] A 0 S
[66] T 5 S [ a(j) ]
[67] [r2:] A 1 S
[68] U 6 S [ a(j+1) ]
[69] S 5 S
[70] E [skip_swap]76 S [ a(j+1) - a(j) >= 0 ? ]
[71] T 0 S
[72] A 5 S [ a(j) ]
[73] [w1:] T 1 S
[74] A 6 S [ a(j+1) ]
```

```
[74] A 6 S [ a(j+1) ]
[75] [w2:] T 0 S

[skip_swap:]
[76] T 0 S

[ обновить инструкции чтения и записи ]
[77] A [cl]122 S
[78] L 0 L
[79] A [rl]65 S
[80] T [rl]65 S
[81] A [cl]122 S
[82] L 0 L
[83] A [r2]67 S
[84] T [r2]67 S
[85] A [cl]122 S
[86] L 0 L
[87] A [wl]73 S
[88] T [wl]73 S
[89] A [cl]122 S
[90] L 0 L
[91] A [w2]75 S
[92] T [w2]75 S

[93] A 2 S
[94] S [cl]122 S
[95] T 2 S [ вычесть 1 из счетчика итераций ]
[96] E [inner_for]59 S [ переход к новой итерации внутреннего цикла ]
[inner_for_exit:]

[97] T 0 S
[98] A [len]124 S
[99] T 2 S [ вернуть счетчик итераций для внутреннего цикла ]
[100] A 1 S
[101] S [cl]122 S
[102] T 1 S [ вычесть 1 из счетчика итераций ]
[ вернуть изначальные инструкции чтения и записи ]
[103] A 10 S
[104] T [rl]65 S
[105] A 11 S
[106] T [r2]67 S
[107] A 12 S
[108] T [wl]73 S
[109] A 13 S
[110] T [w2]75 S
[111] E [outer_for]57 S [ переход к новой итерации внешнего цикла ]
[outer_for_exit:]

[ ответ: (k-1)-й элемент отсортированного массива ]
[112] T 0 S [ очистить аккумулятор ]
[113] A [addr]123 S
[114] A [k]125 S
[115] S [cl]122 S
[116] L 0 L
[117] A [r_ans]119 S
[118] T [r_ans]119 S
[119] [r_ans:] A 0 S
[120] T [res]126 S
[121] Z 0 S
```

```

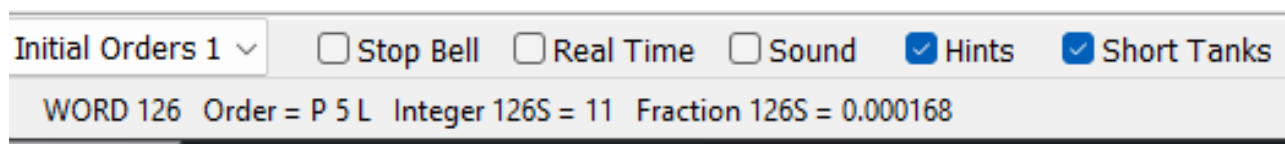
[ блок данных ]
[122] [cl:] P 0 L [const 1]
[123] [addr:] P 63 L [127]
[124] [len:] P 3 S [ длина массива - 2 (для примера 8 - 2 = 6) ]

[125] [k:] P 2 S [4]
[126] [res:] P 0 S [11 для k = 4]

[127] P 5 S [10]
[128] P 7 S [14]
[129] P 8 L [17]
[130] P 2 S [4]
[131] P 6 S [12]
[132] P 5 L [11]
[133] P 7 L [15]
[134] P 4 L [9]
[end:]

```

3.2. Работа программы Orders1



4. Программа Orders2

Алгоритм программы для загрузчика Initial Orders 2 остается таким же. Изменения состоят в том, что Initial Orders 2 поддерживает создание подпрограмм (поэтому этап сортировки массива вынесен в отдельную подпрограмму) и относительную адресацию, из-за чего комментарий перед каждой инструкцией теперь содержит не только абсолютный номер ячейки, но и ее относительный адрес. Главная же подпрограмма, из которой происходит вызов сортировки, предварительно записывает в ячейки 5 и 6 входные данные: адрес сортируемого массива и его длину. После этого вызывается подпрограмма сортировки, а после ее завершения в главной подпрограмме аналогично Orders1 находится a_{k-1} .

Ответ находится в ячейке $\langle ans \rangle$ с адресом 162.

4.1. Код программы Orders2

```
[ Вариант №5 ]
[ Поиск k-й порядковой статистики ]

[ Ячейка для k: 163]
[ Ячейка для ответа: 162]
[ Ячейка для длины массива: 160]
[ Адрес исходного (в последствии - отсортированного) массива: 165]

[
    подпрограмма сортировки массива
]
[ аргументы: длина исходного массива (в ячейке 5) ]
[
    адрес исходного массива (в ячейке 6) ]
[ результат: исходный массив сортируется in-place ]

T 56 K [ установка адреса загрузки ]
GK [ фиксация начального адреса подпрограммы ]
[56.0] A 3F [ формирование кода инструкции возврата ]
[57.1] T [ret] 77@ [ запись инструкции возврата ]

[ записать начальные значения счетчиков для циклов ]
[58.2] A [len] 5F
[59.3] U [outer_iteration_counter] 79@ [ количество оставшихся итераций внешнего цикла ]
[60.4] T [inner_iteration_counter] 80@ [ количество оставшихся итераций внутреннего цикла ]

[ добавить адрес массива для инструкций чтения и записи элементов ]
[61.5] A [addr] 6F
[62.6] A [r1] 29@
[63.7] U [default_r1] 81@
[64.8] T [r1] 29@
[65.9] A [addr] 6F
[66.10] A [r2] 31@
[67.11] U [default_r2] 82@
[68.12] T [r2] 31@
[69.13] A [addr] 6F
[70.14] A [w1] 37@
[71.15] U [default_w1] 83@
[72.16] T [w1] 37@
[73.17] A [addr] 6F
[74.18] A [w2] 39@
[75.19] U [default_w2] 84@
[76.20] T [w2] 39@

[ сортировка ]

[outer_for:]
[77.21] A [outer_iteration_counter] 79@
[78.22] G [outer_for_exit] 76@ [ количество оставшихся итераций < 0 ? ]

[inner_for:]
[79.23] A [inner_iteration_counter] 80@
[80.24] G [inner_for_exit] 61@ [ количество оставшихся итераций < 0 ? ]
[81.25] T 0F
[82.26] A [len] 5F
[83.27] S [inner_iteration_counter] 80@
[84.28] T [j] 85@
[85.29] [r1:] A 0F
[86.30] T [a(j)] 86@
[87.31] [r2:] A 1F
[88.32] U [a(j+1)] 87@
[89.33] S [a(j)] 86@
```



```

[90.34] E [skip_swap] 400 [ a(j+1) - a(j) >= 0 ? ]
[91.35] T 0F
[92.36] A [a(j)] 860
[93.37] [w1:] T 1F
[94.38] A [a(j+1)] 870
[95.39] [w2:] T 0F

[skip_swap:]
[96.40] T 0F

[ обновить инструкции чтения и записи ]
[97.41] A [c1] 780
[98.42] L 0D
[99.43] A [r1] 290
[100.44] T [r1] 290
[101.45] A [c1] 780
[102.46] L 0D
[103.47] A [r2] 310
[104.48] T [r2] 310
[105.49] A [c1] 780
[106.50] L 0D
[107.51] A [w1] 370
[108.52] T [w1] 370
[109.53] A [c1] 780
[110.54] L 0D
[111.55] A [w2] 390
[112.56] T [w2] 390

[113.57] A [inner_iteration_counter] 800
[114.58] S [c1] 780
[115.59] T [inner_iteration_counter] 800 [ вычесть 1 из счетчика итераций ]
[116.60] E [inner_for] 230 [ переход к новой итерации внутреннего цикла ]
[inner_for_exit:]

[117.61] T 0F
[118.62] A [len] 5F
[119.63] T [inner_iteration_counter] 800 [ вернуть счетчик итераций для внутреннего цикла ]
[120.64] A [outer_iteration_counter] 790
[121.65] S [c1] 780
[122.66] T [outer_iteration_counter] 790 [ вычесть 1 из счетчика итераций ]
[ вернуть изначальные инструкции чтения и записи ]
[123.67] A [default_r1] 810
[124.68] T [r1] 290
[125.69] A [default_r2] 820
[126.70] T [r2] 310
[127.71] A [default_w1] 830
[128.72] T [w1] 370
[129.73] A [default_w2] 840
[130.74] T [w2] 390
[131.75] E [outer_for] 210 [ переход к новой итерации внешнего цикла ]
[outer_for_exit:]
[132.76] T 0F
[133.77] [ret:] E 0F

[ зарезервированные ячейки ]
[134.78] [c1:] P 0 D [ const = 1 ]
[135.79] [outer_iteration_counter:] P 0F
[136.80] [inner_iteration_counter:] P 0F
[137.81] [default_r1:] P 0F
[138.82] [default_r2:] P 0F
[139.83] [default_w1:] P 0F
[140.84] [default_w2:] P 0F

```

```

[141.85] [j:] P 0F
[142.86] [a(j):] P 0F
[143.87] [a(j+1):] P 0F

[ конец подпрограммы ]

[ начало программы ]
G K
[144.0] X 0 F
[145.1] A [len] 16 @
[146.2] T 5F [ передача первого параметра ]
[147.3] A [addr] 17 @
[148.4] T 6F [ передача второго параметра ]
[149.5] A 5@ [ \ вызов ]
[150.6] G 56F [ / подпрограммы ]

[151.7] A [k] 19@
[152.8] S [cl] 20@
[153.9] L 0D
[154.10] A [addr] 17@
[155.11] A [r_ans] 13@
[156.12] T [r_ans] 13@
[157.13] [r_ans:] A 0F
[158.14] T [ans] 18@
[159.15] Z 0 F
[ конец программы ]

[160.16] [len:] P 3F [6] [ длина массива - 2 ]
[161.17] [addr:] P 21@ [ адрес массива ]
[162.18] [ans:] P 0F [ ответ ]
[163.19] [k:] P 2F [4]
[164.20] [cl:] P 0D [const = 1]

[ массив ]
[165.21] P 5F [10]
[166.22] P 7F [14]
[167.23] P 8D [17]
[168.24] P 2F [4]
[169.25] P 6F [12]
[170.26] P 5D [11]
[171.27] P 7D [15]
[172.28] P 4D [9]
[end:] EZ PF

```

4.2. Работа программы Orders2

