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 6

Выполнил студент гр. 3530901/00002_____С.А. Колупаев
(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 ячейкипамяти с фиксированными адресами.

Вариант 6: Определение медианы in-place.

1. **Постановка задачи**

Необходимо смоделировать программу для EDSAC, которая определяет медиану неотсортированного массива. Т.е, при четном массиве, программа найдет два числа, которые бы находились ровно посредине уже отсортированного массива и вычислит их сумму, делённую на два. При нечётном же массиве, результат будет равен числу, стоящему посредине отсортированного массива.

Для реализации будем работать с неотсортированным массивом и, при помощи сравнения всех элементов массива друг с другом и счётчиком, вычислять числа/число, находящиеся в середине

2. **Initial Orders 1.**

Программа для EDSAC, реализующая нахождение медианы, использующая загрузчик Initial Orders 1.

```
            T 118[end] S [начало]
            T 5 S  [\]
            T 0 S  [| очищаем рабочие ячейки ]
[count]     T 1 S  [|]
            T 4 S  [/]
            A 112[len] S [\ записываем в раб. ячейку длину массива(i)]
            T 2 S        [/]
            A 112[len] S [\ записываем в раб. ячейку длину массива(j)]
            T 3 S        [/]
            A 114[N0 adr] S [записываем в акк. адрес N0]
            L 0 L [сдвиг на 1 бит влево]
            A 52[<w2>] S [прибавляем к инструкции]
            T 52[<w2>] S [обновляем инструкцию]
[w1]        A 114[N0] S  [записываем в акк. адрес N0]
            L 0 L [сдвиг на 1 бит влево]
            A 58[<w3>] S [прибавляем к инструкции]
            T 58[<w3>] S [обновляем инструкцию]
[loop_for]  A 2 S [помещаем i в акк.]
            S 113[1] S [вычитаем 1]
            G 109[<exit>] S [если результат отрицательный - завершаем программу]
            T 2 S [возвращаем i в раб. ячейку]
[w2]        A 0 S [\запись N в 4 ячейку]
            T 4 S [/]
```

Рис.1. Программа на IO 1 часть 1.

```
53              T 4 S [/]
54  [loop_for2] A 3 S             [\]
55              S 113[1] S        [| уменьшение j с проверкой на положительность]
56              G 78[<end loop>] S [|]
57              T 3 S             [/]
58  [w3]        A 0 S  [запись в аккум M]
59              S 4 S  [M-N]
60              E 66[<f1>] S [проверка результата, если N<M, переходим в 66 ячейку]
61  [f2]        T 0 S [обнуляем аккум]
62              A 1 S       [помещаем в аккум count]
63              A 113[1] S  [добавляем 1]
64              T 1 S       [обновляем count]
65              E 72[<f3>] S [переход к обновлению цикла]
66  [f1]        S 113[1] S [вычитаем 1]
67              G 72[<f3>] S [проверка на N=M]
68              T 0 S [обновляем аккум]
69              A 1 S [помещаем в аккум count]
70              S 113[1] S [вычитаем 1]
71              T 1 S [переход к обновлению цикла]
72  [f3]        T 0 S      [обнуляем аккум]
73              A 113[1] S   [\]
74              L 0 L        [| обновляем инструкцию]
75              A 58[<w3>] S [|]
76              T 58[<w3>] S [/]
77              E 54[<loop_for2>] S [переход в начало цикла]
78  [end loop]  T 0 S [обнуляем аккум]
79              A 1 S [загружаем count]
80              G 86[<f5>] S [проверка count<0]
81              S 113[1] S [вычитаем 1]
82              G 106[<f6>] S [проверка count=0]
83              S 113[1] S [вычитаем 1]
84              G 89[c1] S [проверка count=1]
85              E 95[<f4>] S [иначе топаем в f4]
86  [f5]        A 113[1] S [прибавляем к count 1]
87              E 89[<c1>] S [проверка count=-1]
88              G 95[<f4>] S [иначе топаем в f4]
89  [c1]        T 0 S [обновляем аккум]
90              A 4 S [загружаем N в аккум]
91              R 1 L [делим на 2]
92              A 5 S [прибавляем значения из 5 ячейки]
93              T 5 S [записываем в 5 ячейку]
94              E 95[<f4>] S [топаем в f4]
```

Рис.2. Программа на IO 1 часть 2.

```
94                E 95[<f4>] S [топаем в f4]
95  [f4]          T 0 S [обновляем аккум]
96                A 113[1] S      [\]
97                L 0 L           [| обновляем инструкции]
98                A 52[<w2>] S    [|]
99                T 52[<w2>] S    [|]
100               A 111[ref] S    [|]
101               T 58[<w3>] S    [|]
102               T 1 S           [/]
103               A 112[len] S    [\ обновляем j]
104               T 3 S           [/]
105               E 44[<w1>] S    [начинаем цикл по-новой]
106 [f6]          T 0 S           [обнуляем аккум]
107               A 4 S           [загружаем N в аккум]
108               T 5 S           [записываем N в 5 ячейку]
109 [exit]        O 0 S           [останов]
110               Z 0 S
111 [consts]      A 0 S [ref]     [инструкция для обновления цикла]
112               P 1 L [len]     [длина массива]
113               P 0 L [1]       [1]
114               P 57 L [adr]    [адрес N0]
115               P 4 L           [\]
116               P 8 S           [| массив]
117               P 5 S           [/]
```

Рис.3. Программа на IO 1 часть 3.

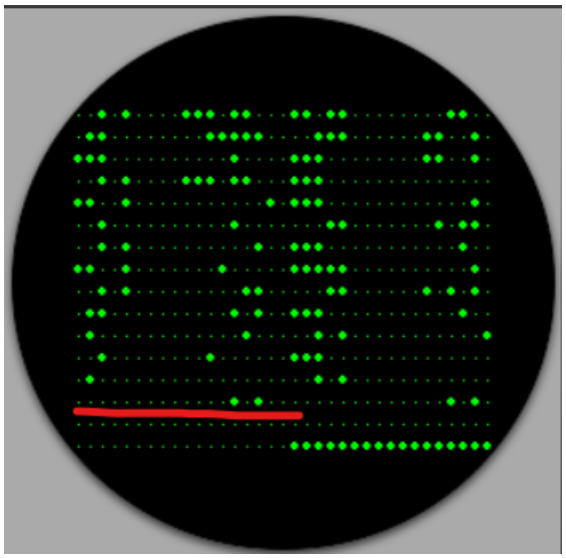


Рис.4. Результат работы программы на IO 1 при нечётном массиве.

WORD 5 Order = P 5 S Integer 5S = 10 Fraction 4L = 0.00015258847

Рис.5. Значение в ячейке 5 в конце работы программы.

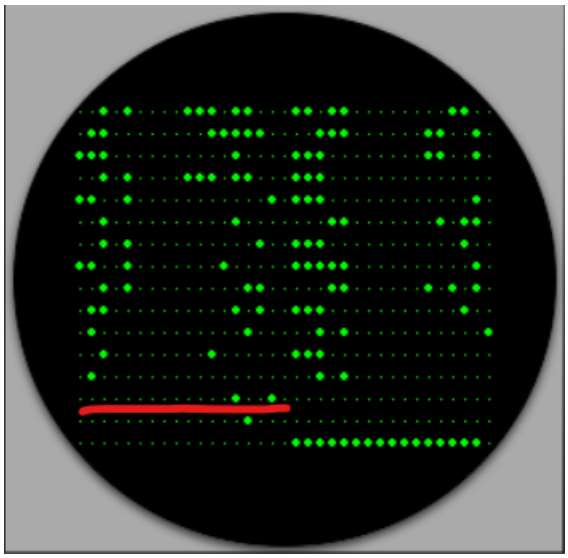


Рис.4. Результат работы программы на IO 1 при чётном массиве.

WORD 5 Order = P 4 L Integer 5S = 9 Fraction 4L = 0.00013732910

Рис.5. Значение в ячейке 5 в конце работы программы.

Рассмотрим более детально работу программы. В начале мы записываем длину массива во 2 и 3 ячейки(это i и j), а также count – счётчик(об его предназначении далее). Далее мы выстраиваем цикл fori{for j{}}, для прохождения по всему массиву.

В этом массиве мы берём, по очереди каждое число и сравниваем его с остальными числами, ориентируясь на их разницу. И вот в зависимости от разницы мы либо прибавляем к count 1(при N>M), либо же её отнимаем(при N<M), также мы никак не изменяем счётчик, если N=M.

Далее мы анализируем на count, если, при чётном массиве, он равен 1 или -1, то его обладатели те самые числа, которые стояли бы посредине в отсортированном массиве, аналогично, если массив нечётный, count = 0 – число, стоящее в середине. Стоит отметить, что count никак не может быть равен 0(при любых числах) в чётном массиве и, аналогично, не может быть равен -1/1 в нечётном. Дальше дело за малым, если массив чётный – делим сумму на два, путём смещения её вправо на 1 бит в двоичном коде, при нечётном, просто записываем число в 5 ячейку. Здесь приведена именно суть работы программы, разбор, непосредственно, кода, вы можете изучить, прочитав комментарии, стоящие напротив каждой строчки, в файле проекта.

3.Initial Orders 2.

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

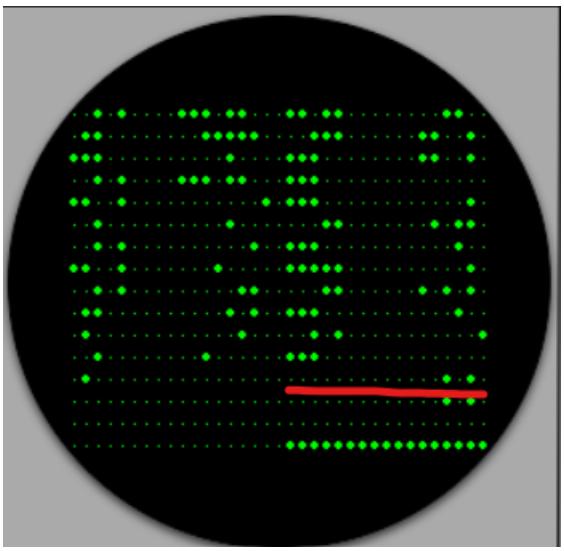
```
T 56[start] K [стартуем]
G K
[0]                     A 3 F  [пролог: формируем код инструкции возврата в Acc]
[1]                     T 71[exit] @ [пролог: записываем инструкцию возврата ]
[2]                     A 7 F   [загружаем адрес N0]
[3]                     A 75[rladr] @ [прибавляем код инструкции]
[4]                     T 14[<w2>] @ [обновляем код инструкции]
[5]                     A 2 F [\ загрузка сформированой интрукции ]
[6]                     T 8 F [/]
[7]  [w1]               A 7 F         [\]
[8]                     A 76[r2adr] @ [| обновления кода инструкции для w3]
[9]                     T 20[<w3>] @  [/]
[10] [loop_for]         A 2 F [помещаем i в акк.]
[11]                    S 73[1] @ [вычитаем 1]
[12]                    G 71[<exit>] @ [если результат отрицательный - завершаем программу]
[13]                    T 2 F [возвращаем i в раб. ячейку]
[14] [w2]               A 0 F [\запись N в 4 ячейку]
[15]                    T 4 @ [/]
[16] [loop_for2]        A 1 F
[17]                    S 73[1] @        [\]
[18]                    G 40[end_loop] @ [| уменьшение j с проверкой на положительность]
[19]                    T 1 F            [|]
[20] [w3]               A 0 F            [/]
[21]                    S 4 @   [запись в аккум M]
[22]                    E 28[<f1>] @ [M-N] [проверка результата, если N<M, переходим в f1]
[23] [f2]               T 0 F [обнуляем аккум]
[24]                    A 5 F [помещаем в аккум count]
[25]                    A 73[1] @ [добавляем 1]
[26]                    T 5 F [обновляем count]
[27]                    E 34[<f3>] @ [переход к обновлению цикла]
[28] [f1]               S 73[1] @ [вычитаем 1]
[29]                    G 34[<f3>] @ [проверка на N=M]
[30]                    T 0 F [обновляем аккум]
[31]                    A 5 F [помещаем в аккум count]
[32]                    S 73[1] @ [вычитаем 1]
[33]                    T 5 F [переход к обновлению цикла]
[34] [f3]               T 0 F [обнуляем аккум]
[35]                    A 73[1] @    [\]
[36]                    L 0 D        [| обновляем инструкцию]
[37]                    A 20[<w3>] @ [|]
[38]                    T 20[<w3>] @ [/]
[39]                    E 16[<loop_for2>] @ [переход в начало цикла]
[40] [end_loop]         T 0 F [обнуляем аккум]
[41]                    A 5 F [загружаем count]
[42]                    G 48[<f5>] @ [проверка count<0]
[43]                    S 73[1] @ [вычитаем 1]
[44]                    G 68[<f6>] @ [проверка count=0]
[45]                    S 73[1] @ [вычитаем 1]
[46]                    G 51[<c1>] @ [проверка count=1]
```

```
[47]                    E 57[<f4>] @ [иначе топаем в f4]
[48] [f5]               A 73[1] @ [прибавляем к count 1]
[49]                    E 51[<c1>] @ [проверка count=-1]
[50]                    G 57[<f4>] @ [иначе топаем в f4]
[51] [c1]               T 0 F [обновляем аккум]
[52]                    A 5 F [загружаем N в аккум]
[53]                    R 1 D [делим на 2]
[54]                    A 6 F [прибавляем значения из 6 ячейки]
[55]                    T 6 F [записываем ответ]
[56]                    E 57[<f4>] @ [топаем в f4]
[57] [f4]               T 0 F [обновляем аккум]
[58]                    A 73[1] @       [\]
[59]                    L 0 D           [|]
[60]                    A 14[<w2>] @    [| обновляем инструкции]
[61]                    T 14[<w2>] @    [|]
[62]                    A 74[adr] @     [/]
[63]                    T 7 F
[64]                    T 5 F
[65]                    A 8 [len] F [\ обновляем j]
[66]                    T 1 F       [/]
[67]                    E 7 [<w1>] @ [начинаем цикл по-новой]
[68] [f6]               T 0 F [обнуляем аккум]
[69]                    A 5 F [загружаем N в аккум]
[70]                    T 6 F [записываем ответ]
[71] [exit]             T 0 F [обнуляем аккум]
[72]                    E 0 F [останов]
[73]                    P 0 D     [1]
[74]                    P 14 F  [adr]
[75]                    A 0 F [r1adr]
[76]                    A 0 F [r2adr]


GK
[0] X 0 F
[1] T 0 F
[2] A 12[adr] @ [адрес массива]
[3] T 7 F       [запись в 7 ячейку]
[4] T 5 F       [обнуление count]
[5] A 13[len] @ [i]
[6] T 2 F
[7] A 13[len] @ [j]
[8] T 1 F
[9] A 9 @  [\ вызов]
[10]G 56 F [/ подпрограммы]
[11]Z 0 F [останов]
[12]P 14 @ [адрес N0]
[13]P 1 D [len]
[array:] [сам массив]
```


```
[array:] [сам массив]
```


```
[14] P 4 D
[15] P 8 F
[16] P 5 F

EZ PF
```

Рис.6. Программа на IO 2.



WORD 6 Order = P 5 F Integer 6F = 10 Fraction 6F = 0.000153

Рис. 6 Результат работы программы на IO2.

5. Вывод

Составленные программы для EDSAC с использованием IO 1 и IO 2 удовлетворяют варианту и находят медиану in-place.