

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

Отчёт по лабораторной работе № 1

Дисциплина: Низкоуровневое программирование

Тема: машина Тьюринга

Выполнил студент гр. 3530901/10003 _____ Я.А. Иванов
(подпись)

Принял старший преподаватель _____ Д.А. Корнеев
(подпись)

“ _ ” _____ 2022 г.

Санкт-Петербург

2022

Оглавление

Техническое задание:.....	3
Метод решения.....	3
ПРОГРАММА ORDERS1.....	5
РАБОТА ПРОГРАММЫ ORDERS1.....	10
ПРОГРАММА ORDERS2.....	11
РАБОТА ПРОГРАММЫ ORDERS2.....	17
Выводы:.....	17

Техническое задание:

Разработать программу для EDSAC, реализующую определение медианы in-place

Программа должна реализовывать заданную функциональность и предполагает загрузчик Initial Orders 1. Массивы данных и другие параметры располагаются в памяти по фиксированным адресам.

Заданную функциональность необходимо выделить в замкнутую подпрограмму, разработать вызывающую её тестовую программу, используя возможности загрузчика Initial Orders 2. Адрес обрабатываемого массива данных и другие параметры передавать через ячейки памяти с фиксированными адресами.

Метод решения

Для поиска медианы нужно выбрать случайно число и расположить по принципу большие - справа, меньшие – слева. Из этих двух групп мы выбираем ту, в которой находится индекс медианы (половина от длины массива). Далее повторяем прошлые шаги с подмассивом до тех пор, пока наше случайное число не будет иметь индекс медианы. Приведём реализацию на языке C

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

int main() {
    int arr[7] = {9, 13, 7, 1, 13, 5, 13};
    int size = sizeof(arr) / sizeof(int);
    int *interval_right = &arr[size];
    int *left = &arr[0];
    int *right;
    int *i;
    int *j;
    int median = (size - 1) / 2;
    int answer;

    while (true) {
        i = left + 1;
        right = interval_right - 1;
        while (i < interval_right) {
            if (*left <= *i) {
                right--;
            }
            i++;
        }
        swap(left, right);
        j = right + 1;
        i = left;
        while (i < right) {
            if (*i >= *right) {
                while (*j >= *right) {
                    j++;
                }
                swap(left, right);
                j++;
            }
            i++;
        }
        if ((int) (right - &arr[0]) == median) {
            answer = *right;
        } else {
            if (median < (right - &arr[0])) {
                interval_right = right;
            } else {
                left = right + 1;
            }
        }
    }
    return 0;
}
```

ПРОГРАММА ORDERS1

[31] T 207 S

[Вариант 4: Определение медианы in-place.]

[ВХОДНЫЕ ДАННЫЕ]

[в ячейке 198 хранится номер начала данных массива, в ячейке 199 хранится длина массив и после идут элементы самого массива]

[массив должен состоять из НЕЧЕТНОГО числа элементов]

[ВЫХОДНЫЕ ДАННЫЕ]

[в ячейке 197 выдается значение медианы]

[КОММЕНТАРИЙ]

[Внутри переменных мы храним не индекс, а команду записи в аккумулятор конкретного элемента]

[PROGRAM]

[32] A 185 [aos:] S [см. ниже]

[33] S 187 [sos:] S [см. ниже]

[34] T 188 [sub_to_add:] S [aos - sos вычисляем значение sub_to_add]

[35] A 185 [aos:] S [см. ниже]

[36] S 186 [tos:] S [см. ниже]

[37] T 189 [add_to_write:] S [aos - tos вычисляем значение sub_to_add]

[38] A 198 [arr:] S [добавляем номер строки начала массива]

[39] A 199 [size:] S [добавляем номер строки начала массива]

[40] S 183 [one:] S

[41] L 0 L [см. ниже]

[42] A 185 [aos:] S [см. ниже]

[43] T 191 [interval_right:] S [сохраним указатель interval_right]

[44] A 198 [arr:] S [добавляем номер строки начала массива]

[45] L 0 L [сдвигаю в разряд индексов]

[46] A 185 [aos:] S [добавляем команду добавления]

[47] T 192 [left:] S [сохраним указатель left]

[48] A 199 [size:] S [загружаем размер]

[49] S 183 [one:] S [прибавляем единицу чтобы округлить в большую сторону]

[50] R 0 L [деление на 2]

[51] T 196 [median:] S [сохранение медианы]

[ЦИКЛ]

[52] [global_loop:] T 15 S [метка глобального - бесконечного цикла]

[53] A 192 [left:] S [загружаем left]

[54] A 184 [index_one:] S [прибавляем единицу к индексу]

[55] T 194 [i:] S [i = left + 1]

[56] A 191 [interval_right:] S [загружаем interval_right]

[57] S 184 [index_one:] S [вычитаем единицу к индексу]

[58] T 193 [right:] S [right = interval_right - 1]

[ЦИКЛ]

[59] [begin_loop_1:] T 15 S [обнуляем аккумулятор]

[60] A 194 [i:] S [плюс i]

[61] S 191 [interval_right:] S [минус interval_right]
[62] E 81 [end_loop_1:] S [ТЕЛО УСЛОВИЯ ЦИКЛА] [while(i меньше чем interval_right) или i - interval_right меньше чем 0]
[63] T 15 S [обнуляем аккумулятор]
[64] A 192 [left:] S
[65] S 188 [sub_to_add:] S
[66] T 70 [var_left_7:] S
[УСЛОВИЕ]
[67] A 194 [i:] S [плюс i]
[68] T 69 [var_i_2:] S
[69] [var_i_2:] A 0 S [для замены i]
[70] [var_left_7:] S 0 S [для замены left]
[71] G 76 [end_if_1:] S [ТЕЛО УСЛОВИЯ] [выполняем условие если left меньше или равен i или i - left больше или равно чем 0]
[72] T 15 S [обнуляем аккумулятор]
[73] A 193 [right:] S [см. ниже]
[74] S 184 [index_one:] S [см. ниже]
[75] T 193 [right:] S [уменьшили right на единицу]
[КОНЕЦ УСЛОВИЯ]
[76] [end_if_1:] T 15 S [обнуляем аккумулятор]
[77] A 194 [i:] S [см. ниже]
[78] A 184 [index_one:] S [см. ниже]
[79] T 194 [i:] S [увеличили right на единицу]
[80] E 59 [begin_loop_1:] S
[81] [end_loop_1:] T 15 S [обнуляем аккумулятор]
[КОНЕЦ ЦИКЛА]
[82] A 193 [right:] S
[83] S 191 [interval_right:] S
[84] G 89 [begin_swap:] S [проверяем что указатель не вышел за пределы. Если вышел, значит его нужно вернуть на крайний элемент массива]
[85] T 15 S [обнуляем аккумулятор]
[86] A 191 [interval_right:] S
[87] S 184 [index_one:] S
[88] T 193 [right:] S
[SWAP]
[89] [begin_swap:] T 15 S
[90] A 192 [left:] S [добавили в аккумулятор]
[91] U 98 [var_left_2:] S
[92] S 189 [add_to_write:] S
[93] T 101 [var_left_3:] S
[94] A 193 [right:] S
[95] U 100 [var_right_2:] S
[96] S 189 [add_to_write:] S
[97] T 103 [var_right_3:] S
[98] [var_left_2:] A 0 S [для замены left]
[99] T 190 [tmp:] S

```

[100] [var_right_2:] A 1 S [для замены right]
[101] [var_left_3:] T 0 S [для замены left]
[102] A 190 [tmp:] S
[103] [var_right_3:] T 1 S [для замены right]
[ENDSWAP]
[104] A 193 [right:] S
[105] S 188 [sub_to_add:] S
[106] U 128 [var_right_5:] S
[107] A 188 [sub_to_add:] S
[108] A 184 [index_one:] S
[109] T 195 [j:] S [j = right + 1]
[110] A 192 [left:] S
[111] U 194 [i:] S [i = left]
[112] T 123 [var_i_1:] S
[113] A 193 [right:] S
[114] S 188 [sub_to_add:] S
[115] T 122 [var_right_4:] S
[ЦИКЛ]
[116] [begin_loop_3:] T 15 S [обнуляем аккумулятор]
[117] A 194 [i:] S
[118] U 123 [var_i_1:] S
[119] S 193 [right:] S
[120] E 159 [end_loop_2:] S [ТЕЛО УСЛОВИЯ ЦИКЛА]
[121] T 15 S [обнуляем аккумулятор]
[УСЛОВИЕ]
[122] [var_right_4:] S 0 S [для замены right]
[123] [var_i_1:] A 0 S [для замены i]
[124] G 154 [end_if_3:] S [ТЕЛО УСЛОВИЯ]
[ЦИКЛ]
[125] [begin_loop_2:] T 15 S [обнуляем аккумулятор]
[126] A 195 [j:] S
[127] T 129 [var_j_1:] S
[128] [var_right_5:] S 0 S [для замены right]
[129] [var_j_1:] A 0 S [для замены j]
[130] G 136 [end_if_2:] S
[131] T 15 S [обнуляем аккумулятор]
[132] A 195 [j:] S
[133] A 184 [index_one:] S
[134] T 195 [j:] S
[135] E 125 [begin_loop_2:] S
[136] [end_if_2:] T 15 S [обнуляем аккумулятор]
[КОНЕЦ ЦИКЛА]
[т.к. в процессе swap нам нужно подменять значения в разных участках кода, была сделата
повторная вставка кода с подметой значений в разных строках]
[SWAP]
[137] A 194 [i:] S [добавили в аккумулятор]

```

[138] U 145 [var_i_5:] S
[139] S 189 [add_to_write:] S
[140] T 148 [var_i_6:] S
[141] A 195 [j:] S
[142] U 147 [var_j_3:] S
[143] S 189 [add_to_write:] S
[144] T 150 [var_j_6:] S
[145] [var_i_5:] A 0 S [для замены left]
[146] T 190 [tmp:] S
[147] [var_j_3:] A 1 S [для замены right]
[148] [var_i_6:] T 0 S [для замены left]
[149] A 190 [tmp:] S
[150] [var_j_6:] T 1 S [для замены right]
[ENDSWAP]
[151] A 195 [j:] S
[152] A 184 [index_one:] S
[153] T 195 [j:] S
[КОНЕЦ УСЛОВИЯ]
[154] [end_if_3:] T 15 S
[155] A 194 [i:] S [см. ниже]
[156] A 184 [index_one:] S [см. ниже]
[157] T 194 [i:] S [сдвигаем указатель к следующему элементу]
[158] E 116 [begin_loop_3:] S
[159] [end_loop_2:] T 15 S [обнуляем аккумулятор]
[КОНЕЦ ЦИКЛА]
[Вычисляем median - right + arr]
[160] S 193 [right:] S
[161] A 185 [aos:] S
[162] R 0 L
[163] A 196 [median:] S
[164] A 198 [arr:] S
[закончили вычисление]
[условие для выражения median _ (right - arr)]
[165] G 179 [smaller_if:] S
[166] S 183 [one:] S
[167] G 173 [equal_if:] S
[168] T 15 S
[median больше чем (right - arr)]
[169] A 193 [right:] S
[170] A 184 [index_one:] S
[171] T 192 [left:] S
[172] E 52 [global_loop:] S [возвращаемся в начало цикла]
[173] [equal_if:] T 15 S [обнуляем аккумулятор] [median равна (right - arr)]
[174] A 193 [right:] S
[175] T 176 [var_right_7:] S
[176] [var_right_7:] A 0 S [для замены right]

[177] T 197 [answer:] S
[178] Z 0 S [СТОП]
[179] [smaller_if:] T 15 S [обнуляем аккумулятор] [median меньше чем (right - arr)]
[180] A 193 [right:] S
[181] T 191 [interval_right:] S
[182] E 52 [global_loop:] S [возвращаемся в начало цикла]
[КОНЕЦ ЦИКЛА]
[CONSANTS]
[183] [one:] P 0 L [единица]
[184] [index_one:] P 1 S [сдвинутая единица]
[185] [aos:] A 0 S [пустой A 0 S]
[186] [tos:] T 0 S [пусой T 0 S]
[187] [sos:] S 0 S [пустной S 0 S]
[188] [sub_to_add:] P 0 S [переменная для превращения S в A] [вычисляется в начале программы]
[189] [add_to_write:] P 0 S [переменная для превращения A в S] [вычисляется в начале программы]
[VARIABLES]
[190] [tmp:] P 0 S
[191] [interval_right:] P 0 S [длина рассматриваемого отрезка]
[192] [left:] P 0 S [левый индекс рассматриваемого отрезка]
[193] [right:] P 0 S [правый индекс рассматриваемого отрезка]
[194] [i:] P 0 S [бегунок 1]
[195] [j:] P 0 S [бегунок 2]
[196] [median:] P 0 S
[OUTPUT]
[197] [answer:] P 0 S [ответ]
[INPUT]
[198] [arr:] P 100 S [номер строки начального элемента массива. В данном примере:]
[199] [size:] P 3 L [размер массива]
[200] P 13 S
[201] P 13 S
[202] P 7 S
[203] P 1 S
[204] P 9 S
[205] P 5 S
[206] P 13 S

РАБОТА ПРОГРАММЫ ORDERS1

WORD 197 Order = P 9 S Integer 197S = 18 F

ПРОГРАММА ORDERS2

Основным отличием загрузчиков Initial Orders 2 от Initial Orders 1 является допустимость подпрограмм. Вынесем код IO1 как подпрограмму и создадим тестовую систему.

Код:

[Вариант 4: Определение медианы in-place.]

[ВХОДНЫЕ ДАННЫЕ]

[в ячейке TODO хранится номер начала данных массива, в ячейке TODO хранится длина массив и после идут элементы самого массива]

[массив должен состоять из НЕЧЕТНОГО числа элементов]

[ВЫХОДНЫЕ ДАННЫЕ]

[в ячейке TODO выдается значение медианы (если кол-во элементов нечетное) или левое число медианы (если кол-во элементов четное)]

[КОММЕНТАРИЙ]

[Внутри переменных мы храним не индекс, а команду записи в аккумулятор конкретного элемента]

[Initial Orders 2 добавляет возможность создания подпрограмм, что позволяет нам вынести @WAP в подпрограмму и разделить программу на логические части, а именно:]

[вычисление констант]

[перенос начального элемента на требуемую позицию]

[перенос больших чисел влево, а меньших - вправо]

[выставление новых границ и проверка на достижение ответа]

[PROGRAM]

GK [поиск медианы]

[44] [0] E 300 F [переход в тестовую систему]

[PROGRAM]

[45] [1] A 154 [abs_aos:] @ [см. ниже]

[46] [2] S 156 [abs_sos:] @ [см. ниже]

[47] [3] T 157 [sub_to_add:] @ [aos - sos вычисляем значение sub_to_add]

[48] [4] A 154 [abs_aos:] @ [см. ниже]

[49] [5] S 155 [abs_tos:] @ [см. ниже]

[50] [6] T 158 [add_to_write:] @ [aos - tos вычисляем значение sub_to_add]

[51] [7] A 167 [arr:] @ [добавляем номер строки начала массива]

[52] [8] A 168 [size:] @ [добавляем номер строки начала массива]

[53] [9] S 152 [one:] @

[54] [10] L 0 D [см. ниже]

[55] [11] A 154 [abs_aos:] @ [см. ниже]

[56] [12] T 160 [interval_right:] @ [сохраним указатель interval_right]

[57] [13] A 167 [arr:] @ [добавляем номер строки начала массива]

[58] [14] L 0 D [сдвигаю в разряд индексов]

[59] [15] A 154 [abs_aos:] @ [добавляем команду добавления]

[60] [16] T 161 [left:] @ [сохраним указатель left]

[61] [17] A 168 [size:] @ [загружаем размер]

[62] [18] S 152 [one:] @ [прибавляем единицу чтобы округлить в большую сторону]

[63] [19] R 0 D [деление на 2]
 [64] [20] T 165 [median:] @ [сохранение медианы]
 [ЦИКЛ]
 [65] [21] [global_loop:] T 15 F [метка глобального - бесконечного цикла]
 [66] [22] A 161 [left:] @ [загружаем left]
 [67] [23] A 153 [index_one:] @ [прибавляем единицу к индексу]
 [68] [24] T 163 [i:] @ [i = left + 1]
 [69] [25] A 160 [interval_right:] @ [загружаем interval_right]
 [70] [26] S 153 [index_one:] @ [вычитаем единицу к индексу]
 [71] [27] T 162 [right:] @ [right = interval_right - 1]
 [ЦИКЛ]
 [72] [28] [begin_loop_1:] T 15 F [обнуляем аккумулятор]
 [73] [29] A 163 [i:] @ [плюс i]
 [74] [30] S 160 [interval_right:] @ [минус interval_right]
 [75] [31] E 50 [end_loop_1:] @ [ТЕЛО УСЛОВИЯ ЦИКЛА] [while(i меньше чем interval_right)
 или i - interval_right меньше чем 0]
 [76] [32] T 15 F [обнуляем аккумулятор]
 [77] [33] A 161 [left:] @
 [78] [34] S 157 [sub_to_add:] @
 [79] [35] T 39 [var_left_7:] @
 [УСЛОВИЕ]
 [80] [36] A 163 [i:] @ [плюс i]
 [81] [37] T 38 [var_i_2:] @
 [82] [38] [var_i_2:] A 0 @ [для замены i]
 [83] [39] [var_left_7:] S 0 @ [для замены left]
 [84] [40] G 45 [end_if_1:] @ [ТЕЛО УСЛОВИЯ] [выполняем условие если left меньше или
 равен i или i - left больше или равно чем 0]
 [85] [41] T 15 F [обнуляем аккумулятор]
 [86] [42] A 162 [right:] @ [см. ниже]
 [87] [43] S 153 [index_one:] @ [см. ниже]
 [88] [44] T 162 [right:] @ [уменьшили right на единицу]
 [КОНЕЦ УСЛОВИЯ]
 [89] [45] [end_if_1:] T 15 F [обнуляем аккумулятор]
 [90] [46] A 163 [i:] @ [см. ниже]
 [91] [47] A 153 [index_one:] @ [см. ниже]
 [92] [48] T 163 [i:] @ [увеличили right на единицу]
 [93] [49] E 28 [begin_loop_1:] @
 [94] [50] [end_loop_1:] T 15 F [обнуляем аккумулятор]
 [КОНЕЦ ЦИКЛА]
 [95] [51] A 162 [right:] @
 [96] [52] S 160 [interval_right:] @
 [97] [53] G 58 [begin_swap:] @ [проверяем что указатель не вышел за пределы. Если вышел,
 значит его нужно вернуть на крайний элемент массива]
 [98] [54] T 15 F [обнуляем аккумулятор]
 [99] [55] A 160 [interval_right:] @
 [100] [56] S 153 [index_one:] @

```

[101] [57] T 162 [right:] @
[SWAP]
[102] [58] [begin_swap:] T 15 F
[103] [59] A 161 [left:] @ [добавили в аккумулятор]
[104] [60] U 67 [var_left_2:] @
[105] [61] S 158 [add_to_write:] @
[106] [62] T 70 [var_left_3:] @
[107] [63] A 162 [right:] @
[108] [64] U 69 [var_right_2:] @
[109] [65] S 158 [add_to_write:] @
[110] [66] T 72 [var_right_3:] @
[111] [67] [var_left_2:] A 0 @ [для замены left]
[112] [68] T 159 [tmp:] @
[113] [69] [var_right_2:] A 1 @ [для замены right]
[114] [70] [var_left_3:] T 0 @ [для замены left]
[115] [71] A 159 [tmp:] @
[116] [72] [var_right_3:] T 1 @ [для замены right]
[ENDSWAP]
[117] [73] A 162 [right:] @
[118] [74] S 157 [sub_to_add:] @
[119] [75] U 97 [var_right_5:] @
[120] [76] A 157 [sub_to_add:] @
[121] [77] A 153 [index_one:] @
[122] [78] T 164 [j:] @ [j = right + 1]
[123] [79] A 161 [left:] @
[124] [80] U 163 [i:] @ [i = left]
[125] [81] T 92 [var_i_1:] @
[126] [82] A 162 [right:] @
[127] [83] S 157 [sub_to_add:] @
[128] [84] T 91 [var_right_4:] @
[ЦИКЛ]
[129] [85] [begin_loop_3:] T 15 F [обнуляем аккумулятор]
[130] [86] A 163 [i:] @
[131] [87] U 92 [var_i_1:] @
[132] [88] S 162 [right:] @
[133] [89] E 128 [end_loop_2:] @ [ТЕЛО УСЛОВИЯ ЦИКЛА]
[134] [90] T 15 F [обнуляем аккумулятор]
[УСЛОВИЕ]
[135] [91] [var_right_4:] S 0 @ [для замены right]
[136] [92] [var_i_1:] A 0 @ [для замены i]
[137] [93] G 123 [end_if_3:] @ [ТЕЛО УСЛОВИЯ]
[ЦИКЛ]
[138] [94] [begin_loop_2:] T 15 F [обнуляем аккумулятор]
[139] [95] A 164 [j:] @
[140] [96] T 98 [var_j_1:] @
[141] [97] [var_right_5:] S 0 @ [для замены right]

```

[142] [98] [var_j_1:] A 0 @ [для замены j]
[143] [99] G 105 [end_if_2:] @
[144] [100] T 15 F [обнуляем аккумулятор]
[145] [101] A 164 [j:] @
[146] [102] A 153 [index_one:] @
[147] [103] T 164 [j:] @
[148] [104] E 94 [begin_loop_2:] @
[149] [105] [end_if_2:] T 15 F [обнуляем аккумулятор]
[КОНЕЦ ЦИКЛА]
[SWAP]
[150] [106] A 163 [i:] @ [добавили в аккумулятор]
[151] [107] U 114 [var_i_5:] @
[152] [108] S 158 [add_to_write:] @
[153] [109] T 117 [var_i_6:] @
[154] [110] A 164 [j:] @
[155] [111] U 116 [var_j_3:] @
[156] [112] S 158 [add_to_write:] @
[157] [113] T 119 [var_j_6:] @
[158] [114] [var_i_5:] A 0 @ [для замены left]
[159] [115] T 159 [tmp:] @
[160] [116] [var_j_3:] A 1 @ [для замены right]
[161] [117] [var_i_6:] T 0 @ [для замены left]
[162] [118] A 159 [tmp:] @
[163] [119] [var_j_6:] T 1 @ [для замены right]
[ENDSWAP]
[164] [120] A 164 [j:] @
[165] [121] A 153 [index_one:] @
[166] [122] T 164 [j:] @
[КОНЕЦ УСЛОВИЯ]
[167] [123] [end_if_3:] T 15 F
[168] [124] A 163 [i:] @ [см. ниже]
[169] [125] A 153 [index_one:] @ [см. ниже]
[170] [126] T 163 [i:] @ [сдвигаем указатель к следующему элементу]
[171] [127] E 85 [begin_loop_3:] @
[172] [128] [end_loop_2:] T 15 F [обнуляем аккумулятор]
[КОНЕЦ ЦИКЛА]
[Вычисляем median - right + arr]
[173] [129] S 162 [right:] @
[174] [130] A 154 [abs_aos:] @
[175] [131] R 0 D
[176] [132] A 165 [median:] @
[177] [133] A 167 [arr:] @
[закончили вычисление]
[условие для выражения median _ (right - arr)]
[178] [134] G 148 [smaller_if:] @
[179] [135] S 152 [one:] @

[180] [136] G 142 [equal_if:] @
 [181] [137] T 15 F
 [median больше чем (right - arr)]
 [182] [138] A 162 [right:] @
 [183] [139] A 153 [index_one:] @
 [184] [140] T 161 [left:] @
 [185] [141] E 21 [global_loop:] @ [возвращаемся в начало цикла]
 [186] [142] [equal_if:] T 15 F [обнуляем аккумулятор] [median равна (right - arr)]
 [187] [143] A 162 [right:] @
 [188] [144] T 145 [var_right_7:] @
 [189] [145] [var_right_7:] A 0 @ [для замены right]
 [190] [146] T 166 [answer:] @
 [191] [147] E 0 D [точка выхода] [для замены]
 [192] [148] [smaller_if:] T 15 F [обнуляем аккумулятор] [median меньше чем (right - arr)]
 [193] [149] A 162 [right:] @
 [194] [150] T 160 [interval_right:] @
 [195] [151] E 21 [global_loop:] @ [возвращаемся в начало цикла]
 [КОНЕЦ ЦИКЛА]
 [CONSANTS]
 [196] [152] [one:] P 0 D [единица]
 [197] [153] [index_one:] P 1 F [сдвинутая единица]
 [198] [154] [abs_aos:] A 0 F [пустой A 0 F]
 [199] [155] [abs_tos:] T 0 F [пусой T 0 F]
 [200] [156] [abs_sos:] S 0 F [пустной S 0 F]
 [201] [157] [sub_to_add:] P 0 F [переменная для превращения @ в A] [вычисляется в начале программы]
 [202] [158] [add_to_write:] P 0 F [переменная для превращения A в @] [вычисляется в начале программы]
 [VARIABLES]
 [203] [159] [tmp:] P 0 F
 [204] [160] [interval_right:] P 0 F [длина рассматриваемого отрезка]
 [205] [161] [left:] P 0 F [левый индекс рассматриваемого отрезка]
 [206] [162] [right:] P 0 F [правый индекс рассматриваемого отрезка]
 [207] [163] [i:] P 0 F [бегунок 1]
 [208] [164] [j:] P 0 F [бегунок 2]
 [209] [165] [median:] P 0 F
 [OUTPUT]
 [210] [166] [answer:] P 0 F [ответ]
 [INPUT]
 [211] [167] [arr:] P 0 F [номер строки начального элемента массива. В данном примере:]
 [212] [168] [size:] P 0 D [размер массива]
 T 300 K
 GK
 [Тест 1]
 [300] A 322 F [загружаем указатель на начало первого массива]
 [301] T 211 F [вставляем в подпрограмму]

[302] A 323 F [загружаем размер]
[303] T 212 F [вставляем в подпрограмму]
[304] A 320 F [загружаем команду для возврата в тестовую систему в аккумулятор]
[305] T 191 F [вставляем в подпрограмму]
[306] E 45 F [переходим в подпрограмму]
[307] A 210 F [точка возврата] [записываем результат в аккумулятор]
[308] T 290 F [сохраняем]
[Тест 2]
[309] A 331 F [загружаем указатель на начало второго массива]
[310] T 211 F [вставляем в подпрограмму]
[311] A 332 F [загружаем размер]
[312] T 212 F [вставляем в подпрограмму]
[313] A 321 F [загружаем команду для возврата в тестовую систему в аккумулятор]
[314] T 191 F [вставляем в подпрограмму]
[315] E 45 F [переходим в подпрограмму]
[316] A 210 F [точка возврата] [записываем результат в аккумулятор]
[317] T 291 F [сохраняем]
[318] ZP [СТОП]
[319] X 0 F
[320] E 307 F [команда для возврата из подпрограммы для первого теста]
[321] E 316 F [команда для возврата из подпрограммы для второго теста]
[INPUTS]
[массив 1]
[322] P 162 F [номер строки начального элемента массива.]
[323] P 3 D [размер массива]
[324] P 14 F
[325] P 15 F
[326] P 7 F
[327] P 1 F
[328] P 9 F
[329] P 5 F
[330] P 13 F
[массив 2]
[331] P 166 D [номер строки начального элемента массива.]
[332] P 2 D [размер массива]
[333] P 10 F
[334] P 10 F
[335] P 10 F
[336] P 10 F
[337] P 10 F
[338] EZPF

РАБОТА ПРОГРАММЫ ORDERS2

В данном случае мы сохраняем результаты тестов в ячейки 290 и 291:

```
WORD 290  Order = P 9 F  Integer 290F = 18
```

```
WORD 291  Order = P 10 F  Integer 291F = 20
```

Выводы:

В ходе выполнения данной лабораторной работы была разработана программа для EDSAC для поиска медианы массива.