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

**Отчёт по лабораторной работе № 2**

Дисциплина: Низкоуровневое программирование

Тема: Программирование на Edsac  
Вариант: 13

Выполнил студент гр. 3530901/10003 Фаретдинов Р.А  
 Принял преподаватель Коренев Д.А

Санкт-Петербург  
2022

**1. ТЗ………………………………………………………..стр.3**

**2. Метод решения………………………………………...стр.3**

**3. Программа Orders 1………………………………….стр.3**

**4. Работа программы Orders 1……………………………………стр.4-5**

**5.Программа Orders 2…………………………………………**

**6.Работа программы Orders 2………………………………..**

1. **ТЗ**  
   Найти в массиве минимальный элемент
2. **Метод решения**

программа принимает на вход:

адрес начала массива (ячейка 62)

длину массива (ячейка 63)

и соответственно сам массив (ячейки 67 - 76)

минимальный элемент (66)

Пусть у нас есть массив длинной N, индексация которого начинается с единицы, тогда , чтобы найти наименьший элемент надо сравнить каждый последующий элемент массива с предыдущим, записывая текущий минимальный элемент в min. Изначально 1 элемент массива сравнивается с большим числом, например 20.

Пример

Рассмотрим массив, состоящий из чисел 7, 6, 3, 4. Возьмем счетчик, равный длине массива. Перед каждой итерацией будем уменьшать его на 1, и если он меньше 0 программа будет переходить к завершению.

Начало: 7 6 3 4 счетчик =

Проверка условий перед первой итерации: счетчик = 4-1=3>0

1-ая итерация:

текущий элемент – temp: 7

min:20

если текущий элемент меньше текущего минимального элемента, то минимальный элемент обновляется

7<20 => min:7

Проверка условий перед 2-ой итерации: счетчик = 3-1=2>0

2-ая итерация:

temp: 6

Min:7

6<7 => min = 6

Проверка условий перед 3-ей итерацией: счетчик = 2-1=1>0

3-ая итерация:

Temp: 3

Min: 6

3<6 => min=3

Проверка условий перед последней итерацией: счетчик = 1-1 = 0 =0

4-ая итерация:

Temp: 4

Min: 3

4> 3 min=3

Проверка перед 5-ой итерацией: счетчик = 0-1 = -1< 0 , завершение программы.

1. **Программа Orders 1**

[31] T 77 S

**Подготовка к итерации**

[32] [loop:] A 62 [<start>] S [загрузка в аккумулятор адреса элемента]

[33] L 0 L [сдвиг аккумулятора на 1 влево]

[34] A 60 [<mc1>] S [прибавка к аккумулятору инструкции с нулевым адресом]

[35] T 44 [<m1>] S [запись инструкции в рабочую ячейку]

[36] A 62 [<start>] S [загрузка в аккумулятор адреса элемента]

[37] L 0 L [сдвиг аккумулятора на 1 влево]

[38] A 61 [<mc2>] S [прибавка к аккумулятору инструкции с нулевым адресом]

[39] T 46 [<m2>] S [запись инструкции в рабочую ячейку]

[40] A 63 [<len>] S [загрузка счётчика]

[41] S 64 [<c1>] S [уменьшение счётчика на 1]

[42] G 59 [<final>] S [если счётчик меньше нуля, то переходит к завершению]

[43] T 63 [<len>] S [возврат счётчика]

**Основное тело цикла**

[44] [m1:] A 0 S [забор первого элемента пары]

[45] U 1 [<temp>] S [отправление его в temp]

[46] [m2:] T 0 S [добавление элемента]

[47] A 1[temp] S [загрузка текущего элемента на акк]

[48] S 66 [res]S [вычитание акк - res]

[49] G 51 S [если акк - res < 0, go to 51]

[50] E 54 S [если акк - res >= 0, go to 54]

[51] T 0 S [обнуление акк]

[52] A 1 [temp]S [загрузка temp на акк]

[53] T 66[res] S [записываем в res значение акк(temp)]

[54] T 1[<temp>] S [обнуление temp]

[55] A 62 [<start>] S [забирает адрес элемента]

[56] A 64 [<c1>] S [увеличивает его на 1]

[57] T 62 [<start>] S [возвращает на место]

[58] E 32 [<loop>] S [переход в начало цикла]

[59] [final:]Z 0 S [завершение программы]

**Константы и входные данные**

[60] [mc1:] A 0 S [шаблон для забора числа из нужной ячейки]

[61] [mc2:] T 0 S [шаблон для добавления числа в нужную ячейку]

[62] [start:] P 33 L [адрес первого элемента]

[63] [len:] P 5 S [длина массива]

[64] [c1:] P 0 L [константа = 1]

[65] [c2:] P 1 S [константа = 2]

[66] [res] P 10 S [min]

[67] [aryay:] P 2 L [5]

[68] P 4 S [8]

[69] P 1 L [3]

[70] P 2 S [4]

[71] P 2 L [5]

[72] P 3 S [6]

[73] P 3 L [7]

[74] P 4 S [8]

[75] P 4 L [9]

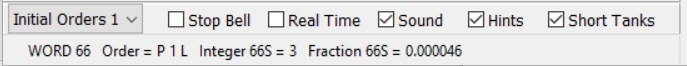
[76] P 5 S [10]

1. **Работа программы Orders 1**

Из записи программы мы знаем, что массив состоит из следующих элементов: 5 8 3 4 5 6 7 8 9 10

Минимальным элементом массива является – 3

Результат записывается в ячейку 66



1. **Программа Orders 2**

[54] T 56 K [директива IO2, установка адреса загрузки]

[55] G K [директива IO2, фиксация начального адреса подпрограммы]

[56] [0]A 3 F [пролог: формирование инструкции возврата в Acc]

[57] [1]T 24 [return] @ [пролог: запись инструкции возврата]

**Подготовка к итерации:**

[58] [2]A 0 F [загрузка в аккумулятор адреса 0гог элемента]

[59] [3]A 26 [mc1] @ [прибавка к аккумулятору инструкции с нулевым адресом]

[60] [4]T 9 [m1] @ [запись инструкции в рабочую ячейку]

[61] [5]A 2 [loop] F [загрузка счетчика]

[62] [6]S 25 [c1] @ [уменьшаем его на 1]

[63] [7]G 23 [exit] @ [если результат <0 - выходим их программы]

[64] [8]T 2 F [возврат счетчика]

**Основное тело цикла**

[65] [9]A 0 [m1] F [забор первого элемента]

[66] [10]T 1 F [отправление его в temp]

[67] [11]A 1 [temp]F [загрузка текущего элемента из temp]

[68] [12]S 4 [min]F [вычитаем текущий наименьший элемент]

[69] [13]G 15 @ [если <0 - прибавляем текущий наибольший элемент]

[70] [14]E 17 @ [если >=0 обнуляем акк]

[71] [15]A 4 [min]F

[72] [16]T 4 F [текущий элемент становится наименьшим на данный момент]

[73] [17]T 3 F [обнуление]

[74] [18]A 25 [c1] @ [загрузка в акк единицы]

[75] [19] L 0 D [сдвигаем на 1 разряд влево]

[76] [20] A 9 @ [прибавляем код инструкции, исполненной на предыдущем шаге]

[77] [21] T 9 @ [запись сформированной инструкции, обнуление аккумулятора]

[78] [22] E 5 [loop] @ [переход в начало цикла]

**Завершение подпрограммы**

[79] [23] T 0 F [exit] [обнуление]

[80] [24] E 0 F [return] [Инструкция выхода из программы]

**Константы подпрограммы**

[81] [25] P 0 D [c1] [const = 1]

[82] [26] A 0 [mc1]F [основа для формирования инструкции с меткой m1]

**Начало основной программы**

[83] G K [директива IO2, фиксация начального адреса программы]

**Перемещение входных данный в ячейки с 0-ой по 4-ую для удобства**

[84] [0]A 9 [len] @ [адресс масива ]

[85] [1]T 0 [count] F [её запись в ячейку 0]

[86] [2]A 10 [start] @ [длинна массива]

[87] [3]T 2 F [его запись в ячейку 2]

[88] [4]A 20 [res] @ [минимальный элемент]

[89] [5]T 4 F [его запись в ячейку 4]

[90] [6]A 6 F [\ вызов]

[91] [7]G 56 F [/ подрограммы] [запись res в ячейку 3]

[92] [8]Z 0 F [остановка]

**Входные данные**

[93] [9] P 11 [start] @ [адрес начала массива]

[94] [10] P 4 [len] D [длина массива]

[95] [11] P 3 D [array] [7]

[96] [12]P 3 F [6]

[97] [13]P 2 D [5]

[98] [14]P 2 F [4]

[99] [15]P 1 D [3]

[100] [16]P 3 D [7]

[101] [17]P 4 F [8]

[102] [18]P 4 D [9]

[103] [19]P 5 F [10]

[104] [20] P 10 F [res]

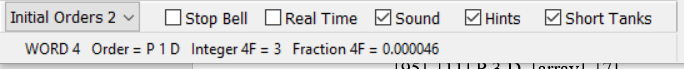
[105] EZ PF [директива IO2, переход к исполнению]

1. **Работа программы Orders 2**

Из записи программы мы знаем, что массив состоит из следующих элементов: 5 8 3 4 5 6 7 8 9 10

Минимальным элементом массива является – 3

Результат записывается в ячейку: 4



Директива GK позволяетсделать текст «шаблона» независимым от начального адреса загрузки формируемого фрагмента кода, воспользовавшись кодом “@” , например:

[83] Gk [@=Mem[83]]

[84] [@+0]A 9 [len] @ [адресс масива ]

[85] [@+1]T 0 [count] F [её запись в ячейку 0]

[86] [@+2]A 10 [start] @ [длинна массива]

[87] [@+3]T 2 F [его запись в ячейку 2]

[88] [@+4]A 20 [res] @ [минимальный элемент]

[89] [@+5]T 4 F [его запись в ячейку 4]

[90] [@+6]A 6 F [\ вызов]

поэтому массив находится в диапазоне ячеек (11-19).

**Результат программы я записываю в ячейку 4.**

Изначально я беру большое число и записываю его в ячейку

[104] [20] P 10 F [res],

Затем я загружаю значение с этой ячейки в acc и в последствии записываю в ячейку 4, для удобного обращения к минимальному элементу в подпрограмме.

[84] [0]A 9 [len] @ [адресс масива ]

[85] [1]T 0 [count] F [её запись в ячейку 0]

[86] [2]A 10 [start] @ [длинна массива]

[87] [3]T 2 F [его запись в ячейку 2]

[88] [4]A 20 [res] @ [минимальный элемент]

[89] [5]T 4 F [его запись в ячейку 4]