

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

Отчёт по лабораторной работе

Дисциплина: Низкоуровневое программирование

Тема: EDSAC

Выполнил студент гр. 3530901/10005
Калашников О. Ю.

(подпись)

Преподаватель
Коренев Д.А.

(подпись)

“ ____ ” _____ 2022 г.

Санкт-Петербург
2022

Оглавление

ТЗ	3
Метод решения.....	3
Программа Initial Orders 1	3
Пояснение к программе	3
Код программы.....	4
Результат работы программы	5
Программа Initial Orders 2	5
Пояснение к программе	5
Код программы.....	6
Результат работы программы	7
Вывод	7

ТЗ

Написать программу для машины EDSAC, которая реализует нахождение максимального элемента в массиве чисел. Необходимо реализовать эту программу для обоих загрузчиков: Initial Orders 1 и 2.

Метод решения

Для нахождения максимального элемента массива необходимо циклически пройти каждый элемент, сравнивая его с максимальным значением, найденным на данный момент (для первого элемента текущее максимальное значение принимается за 0).

Например, рассмотрим массив [3, 5, 9, 4, 1]:

1. $3 > 0 \Rightarrow 3$ — текущий максимальный элемент;
2. $5 > 3 \Rightarrow 5$ — текущий максимальный элемент;
3. $9 > 5 \Rightarrow 9$ — текущий максимальный элемент;
4. $4 < 9 \Rightarrow 9$ — остаётся текущим максимальным элементом;
5. $1 < 9 \Rightarrow 9$ — остаётся текущим максимальным элементом;

Ответ: 9.

Программа Initial Orders 1

Пояснение к программе

Программа начинается с 31-й ячейки, так как все предыдущие ячейки используются для работы загрузчика, но после начала программы их также можно использовать. Ячейки с 59-й по 63-ю используются для хранения вспомогательных значений:

59 — хранит текущее максимальное значение (сначала 0), а после выполнения программы содержит ответ;

60 — константная единица;

61 — длина массива без одного элемента, так как команды EDSACa не позволяют делать проверку меньше или равно нулю, а только меньше нуля;

62 — значение ячейки, где хранится первый элемент массива;

63 — временно хранит текущий элемент массива.

В ячейках 65-73 хранятся значения элементов массива.

На каждой итерации массива программа сначала записывает текущий элемент массива во временное хранилище (ячейка 63). Затем при помощи вычитания максимального элемента из текущего элемента, проверяет, является ли этот новый элемент максимальным на данный момент. Далее программа перезаписывает инструкцию получения элемента массива, чтобы на следующей итерации записать новый элемент. При перезаписывании инструкции используется команда LOL (сдвиг значения аккумулятора на один разряд влево) — это нужно по той причине, что поле адреса инструкции в EDSAC начинается не с 0-го разряда, а с 1-го. В конце программы из ячейки длины массива вычитается 1, так как мы прошли элемент, а затем проверяется, не стала ли длина массива отрицательной. Если стала, то программа завершается, если нет, то начинается новая итерация цикла.

Код программы

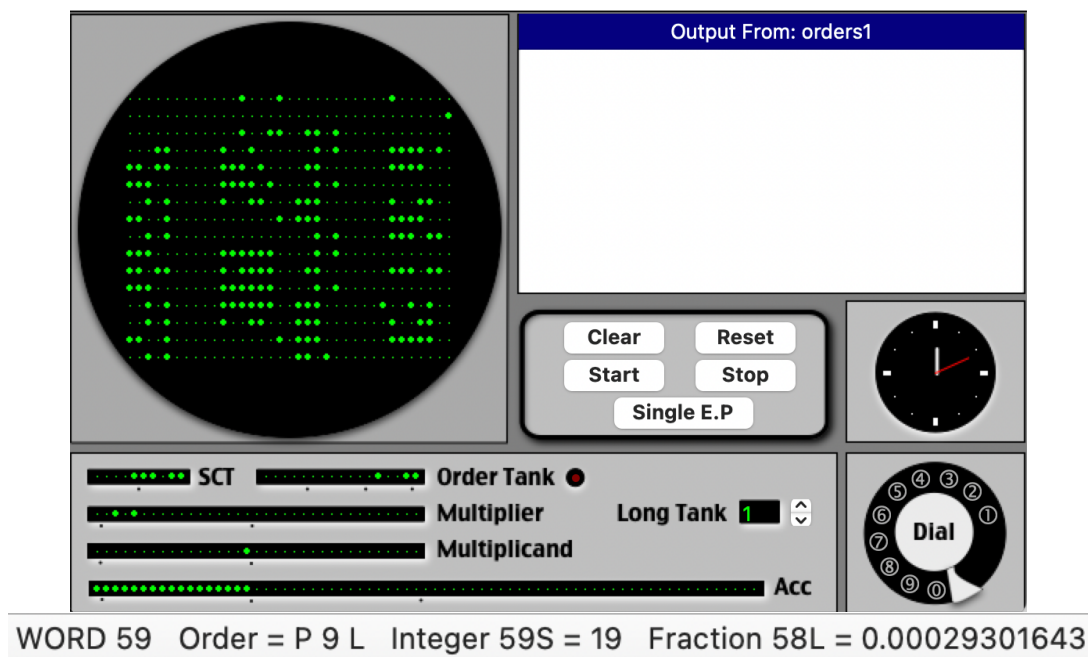
```

29 [ Номер задания: 12 — максимальный элемент массива ]
30 [ Ячейка с ответом: 59 ]
31 T74S [ последняя ячейка ]
32 X0S [ старт ]
33 T0S [ очищение асс ]
34 A62S [ добавление в асс номера ячейки первого элемента массива ]
35 LOL [ сдвиг влево ]
36 A38[<r>]S [ прибавление кода инструкции с нулевым адресом ]
37 T38[<r>]S [ запись сформированной инструкции и обнуление асс ]
38 [r:] A0S [ инструкция для загрузки в асс значения ячейки ]
39 T63S [ запись значения элемента массива в ячейку 63 ]
40 T0S [ очищение асс ]
41 A63S [ добавление в асс значения текущего элемента массива ]
42 S59S [ отнимаем от асс текущий максимальный элемент массива ]
43 G47S [ если новый элемент меньше, то пропускаем этап записи в результат ]
44 T0S [ очищение асс ]
45 A63S [ добавление в асс значение нового максимального элемента ]
46 T59S [ добавление в ячейку с результатом ]
47 T0S [ очищение асс ]
48 A60S [ добавление в асс константу единицу ]
49 LOL [ сдвиг влево ]
50 A38S [ добавление инструкции для получения предыдущего элемента массива ]
51 T38S [ записывает инструкцию для следующего элемента в ячейку 38 ]
52 T0S [ очищаем асс ]
53 A61S [ добавление в асс длины массива ]
54 S60S [ отнимаем единицу, так как прошли один элемент ]
55 G58S [ если длина меньше 0, то завершаем программу ]
56 T61S [ записывает новую длину в ячейку 61 ]
57 E36S [ стартуем новую итерацию цикла ]
58 Z0S [ стоп ]
59 P0S [зарезервированная ячейка под ответ]
60 P0L [константа единица]
61 P4L [ длина массива без одного элемента (10 - 1) ]
62 P32S [ значение ячейки первого элемента массива ]
63 P0S [ зарезервированная ячейка для временного хранения текущего элемента]
64 [начало массива] P 8 S [ 16 ]
65 P 8 S [ 16 ]
66 P 1 S [ 2 ]
67 P 7 L [ 15 ]
68 P 9 L [ 19 (максимальное) ]
69 P 3 L [ 7 ]
70 P 3 S [ 6 ]
71 P 4 L [ 9 ]
72 P 8 L [ 17 ]
73 P 8 L [ 17 ] [ конец массива ]

```

Результат работы программы

Возьмём массив [16, 16, 2, 15, 19, 7, 6, 9, 17, 17]. Ожидаемый результат — 19.



Результат совпал с ожидаемым.

Программа Initial Orders 2

Пояснение к программе

При написании кода для Initial Orders 2 программа делится на две части: подпрограмма и программа, её вызывающая. В Initial Orders 2 используется относительная адресация — это значит, что адреса ячеек отсчитываются не с первой строчки программы, а с начального адреса подпрограммы (в моём случае 56). Адреса ячеек с учётом относительной адресации указаны в комментариях в формате <адрес по относительной адресации>.<адрес по локальной адресации> — именно поэтому несмотря на то, что ячейка с результатом находится на строчке 34, её фактическое положение будет в 84-й ячейке.

Логика работы программы, набор констант и зарезервированных значений остаются точно такими же, как и в Initial Orders 1. Изменения состоят только в том, что для обращения к ячейке или записи значения в

неё по локальной адресации используется код длины слова “@”, а для относительной адресации — «F». Также в Initial Orders 2 другой формат записи констант: D вместо L и F вместо S.

Код программы

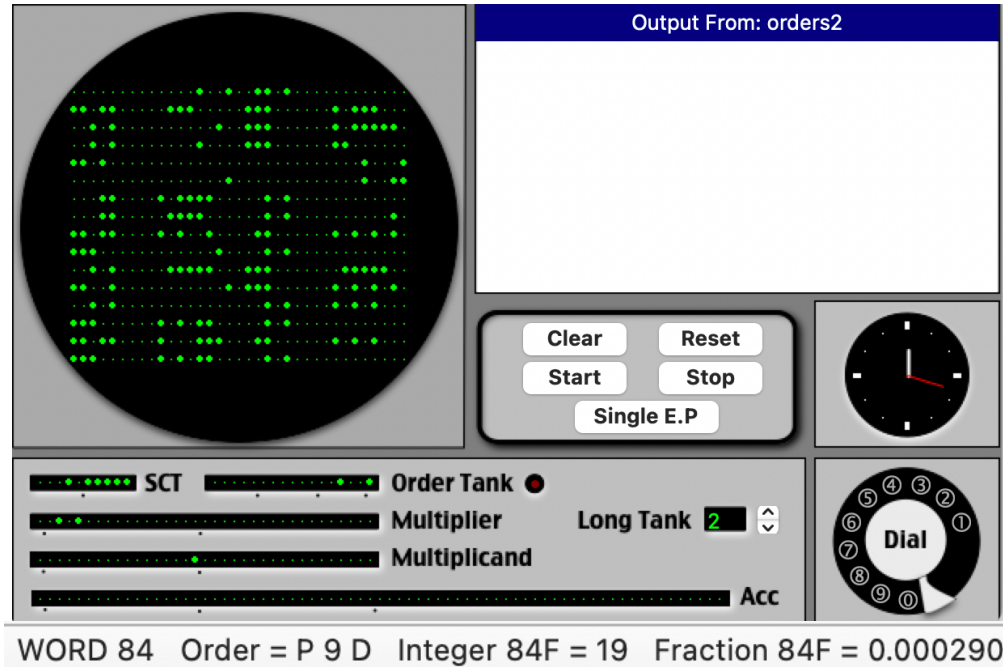
```

1  [ Номер задания: 12 — максимальный элемент массива ]
2  [ Ячейка с ответом: 84 ]
3  T56K [ задаём адрес подпрограммы ]
4  G K  [ фиксируем адрес подпрограммы ]
5  [ Начало подпрограммы ]
6  [56.0]A3F [ формируем инструкцию возврата ]
7  [57.1]T27@ [ записываем инструкцию возврата ]
8  [58.2]T0F [ очищение асс ]
9  [59.3]A4F [ добавление в асс номера ячейки первого элемента массива ]
10 [60.4]A6[<r>]@ [ прибавление кода инструкции с нулевым адресом ]
11 [61.5]T6[<r>]@ [ запись сформированной инструкции и обнуление асс ]
12 [62.6] [r:] A0F [ инструкция для загрузки в асс значения ячейки ]
13 [63.7]T30@ [ запись значения элемента массива во временное хранилище ]
14 [64.8]T0F [ очищение асс ]
15 [65.9]A30@ [ добавление в асс значения текущего элемента массива ]
16 [66.10]S28@ [ отнимаем от асс текущий максимальный элемент массива ]
17 [67.11]G15@ [ если новый элемент меньше, то пропускаем этап записи в результат ]
18 [68.12]T0F [ очищение асс ]
19 [69.13]A30@ [ добавление в асс значение нового максимального элемента ]
20 [70.14]T28@ [ добавление в ячейку с результатом ]
21 [71.15]T0F [ очищение асс ]
22 [72.16]A29@ [ добавление в асс константу единицу ]
23 [73.17]L0D [ сдвиг влево ]
24 [74.18]A6@ [ добавление инструкции для получения предыдущего элемента массива ]
25 [75.19]T6@ [ записывает инструкцию для следующего элемента ]
26 [76.20]T0F [ очищаем асс ]
27 [77.21]A1F [ добавление в асс длины массива ]
28 [78.22]S29@ [ отнимаем единицу, так как прошли один элемент ]
29 [79.23]G26@ [ если длина меньше 0, то завершаем программу ]
30 [80.24]T1F [ записывает новую длину ]
31 [81.25]E4@ [ стартуем новую итерацию цикла ]
32 [82.26]T0F [ оочищение асс ]
33 [83.27]E0F [ инструкция возврата из подпрограммы ]
34 [84.28]P0F [зарезервированная ячейка под ответ]
35 [85.29]P0D [константа единица]
36 [86.30]P0F [ зарезервированная ячейка для временного хранения текущего элемента]
37
38 [ Начало программы ]
39 G K [ фиксируем адрес программы ]
40 [87.0] X0F [ старт ]
41 [88.1] A9@ [ добавляем в асс адрес первого элемента массива ]
42 [89.2] T4F [ запись адреса первого элемента массива в ячейку 4 ]
43 [90.3] A8@ [ запись в асс длины массива ]
44 [91.4] T1F [ запись длины массива в ячейку 1 ]
45 [92.5] A5@ [ вызов ]
46 [93.6] G56F [ подпрограммы ]
47 [94.7] Z0F [ стоп ]
48 [95.8]P4D [ длина массива без одного элемента (10 - 1) ]
49 [96.9]P10@ [ значение ячейки первого элемента массива ]
50
51 [ Начало массива ]
52 [97.10]P 8 F [ 16 ]
53 [98.11]P 8 D [ 17 ]
54 [99.12]P 1 F [ 2 ]
55 [100.13]P 7 D [ 15 ]
56 [101.14]P 9 D [ 19 (максимальное) ]
57 [102.15]P 3 D [ 7 ]
58 [103.16]P 3 F [ 6 ]
59 [104.17]P 4 D [ 9 ]
60 [105.18]P 8 D [ 17 ]
61 [106.19]P 8 D [ 17 ]
62 [ Конец массива ]
63 EZ PF

```

Результат работы программы

Возьмём массив [16, 17, 2, 15, 19, 7, 6, 9, 17, 17]. Ожидаемый результат — 19.



Результат совпал с ожидаемым.

Вывод

В ходе выполнения лабораторной работы мной была разработана программа для машины EDSAC для обоих вариантов загрузчиков Initial Orders 1 и 2, которая реализует нахождение максимального элемента в заданном массиве чисел. Результаты работы программы для обоих загрузчиков совпали с ожидаемыми.