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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Минин П.В.

(подпись)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Коренев Д.А.

(подпись)

Санкт-Петербург

2022

**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**

Чтобы найти минимальное значение массива необходимо пройтись по всем элементам массива и сравнить их с минимальным на данный момент значением. Для первого элемента массива текущее минимальное значение принимается равным 58982 (fraction = 0.9).

Для примера рассмотрим массив [7,4,5,2,9].

1. 58982 > 7 (7 текущий минимальный элемент)
2. 7 > 4 (4 текущий минимальный элемент)
3. 4 < 5 (4 текущий минимальный элемент)
4. 4 > 2 (2 текущий минимальный элемент)
5. 2 < 9 (2 текущий минимальный элемент)

В итоге доходим до конца массива и получаем, что минимальный элемент равен 2.

**Программа Initial Orders 1**

Программа начинается с 31 ячейки, так как первые тридцать ячеек используются для работы загрузчика, но после запуска программы их можно использовать. Ячейки с 37 по 53 – это цикл, который нужен для прохождения по всем элементам массива и сравнения каждого элемента с минимальным на данный момент значением. В начале каждой итерации происходит запись очередного элемента массива во временное хранилище (ячейка 58), затем из текущего элемента вычитаем минимальный на данный момент (ячейка 56). Если получилось отрицательное число, то записываем новое минимальное значение. Далее перезаписываем инструкцию для добавления очередного элемента массива в начале итерации. Проверяем не дошли ли мы до конца массива, для этого вычитаем из текущей длины массива (ячейка 55) единицу (ячейка 57) и если получилось число отрицательное, то заканчиваем выполнение программы, иначе записываем новую длину массива и переходим к следующей итерации цикла. Так же стоит сказать что размер массива берется на один меньше, так как есть лишь проверки на положительное и отрицательное число.

Ячейки с 55 по 58 используются для хранения вспомогательных значений:

55 – длина массива без одного элемента (n - 1).

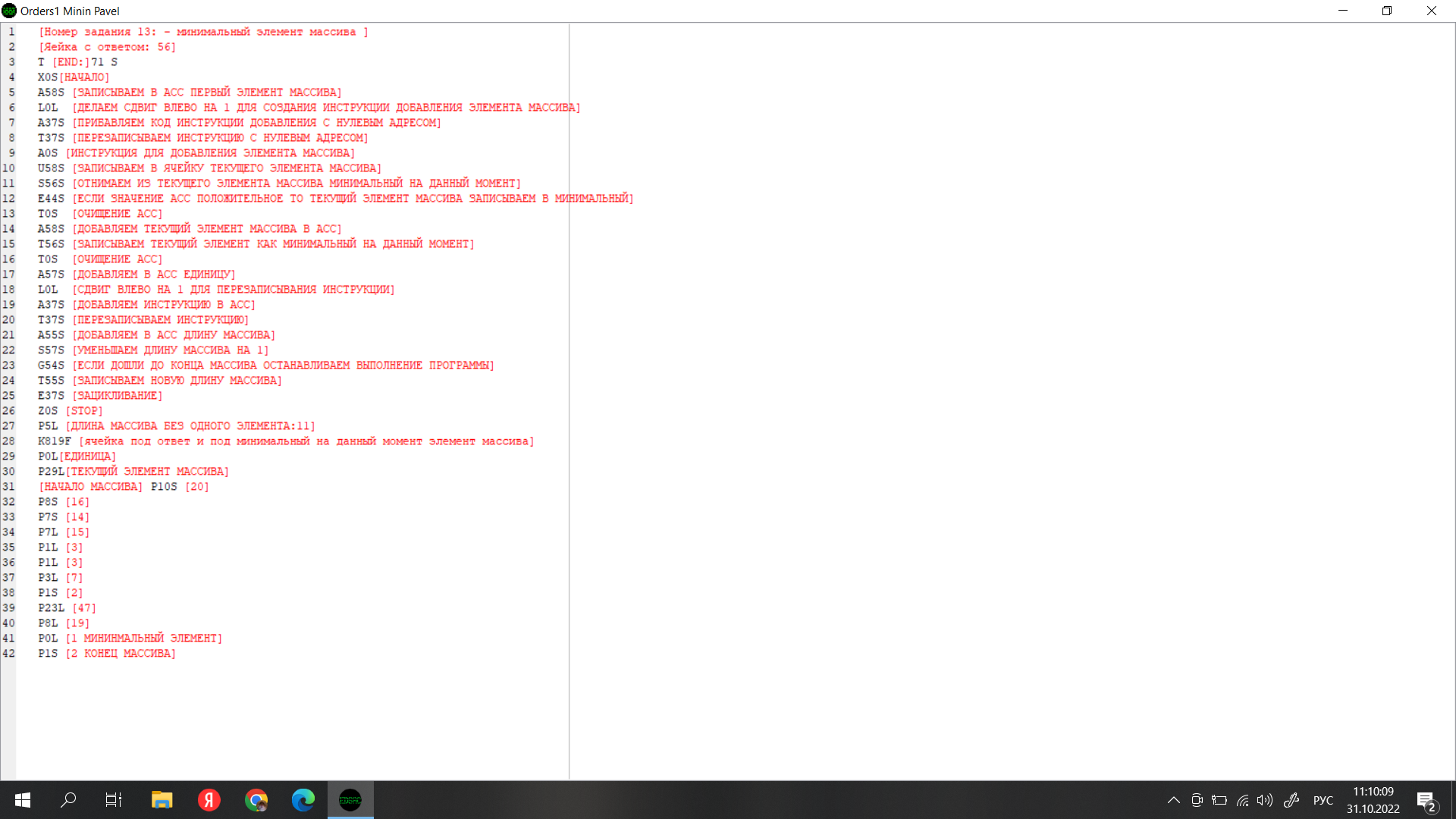
56 – ячейка с минимальным на данный момент элементом массива, в конце программы в ней будет лежать ответ.

57 – ячейка в которой хранится единица

58 – ячейка в которой хранится текущий элемент массива

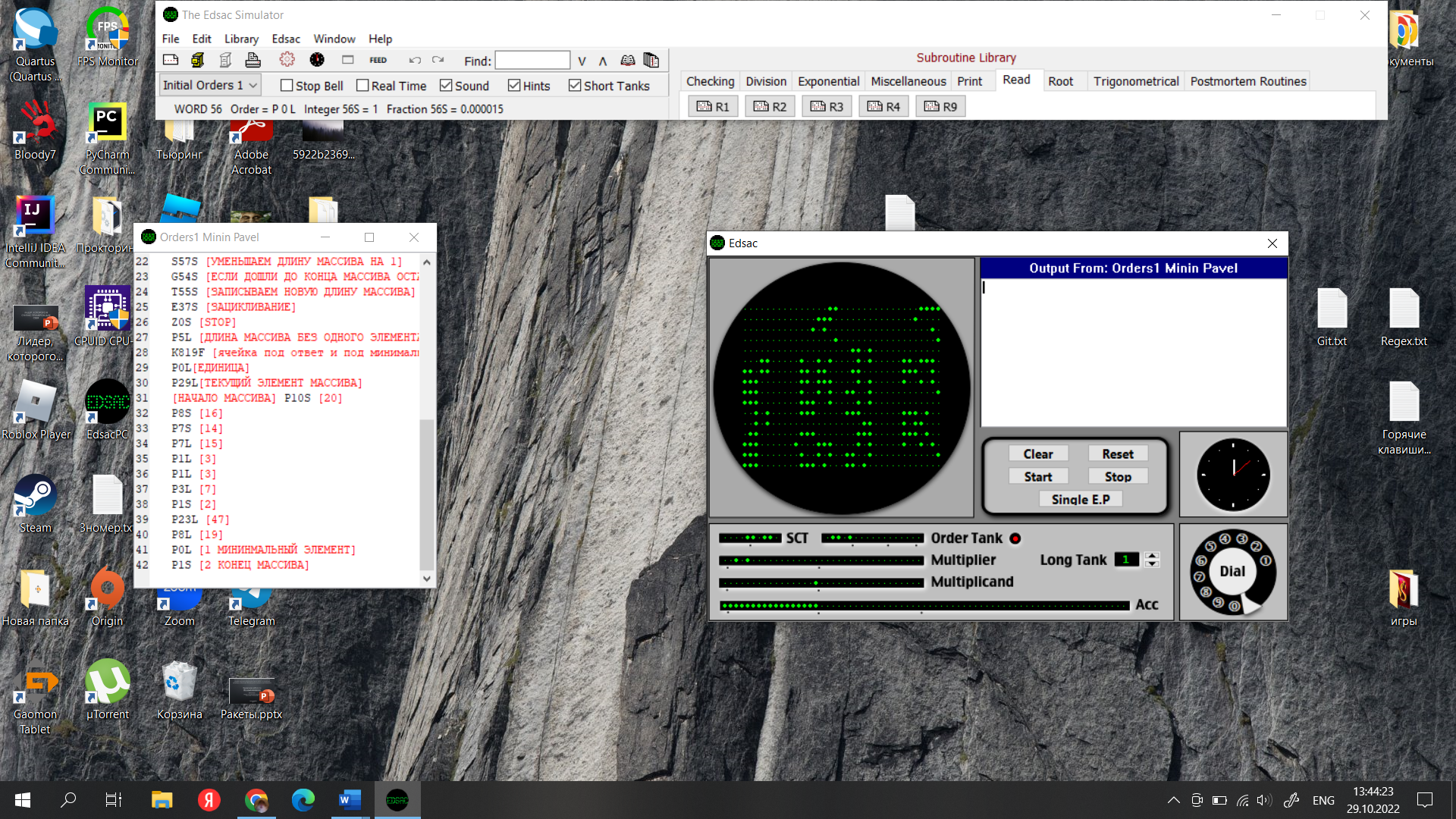
Ячейки с 59 хранят в себе элементы массива.

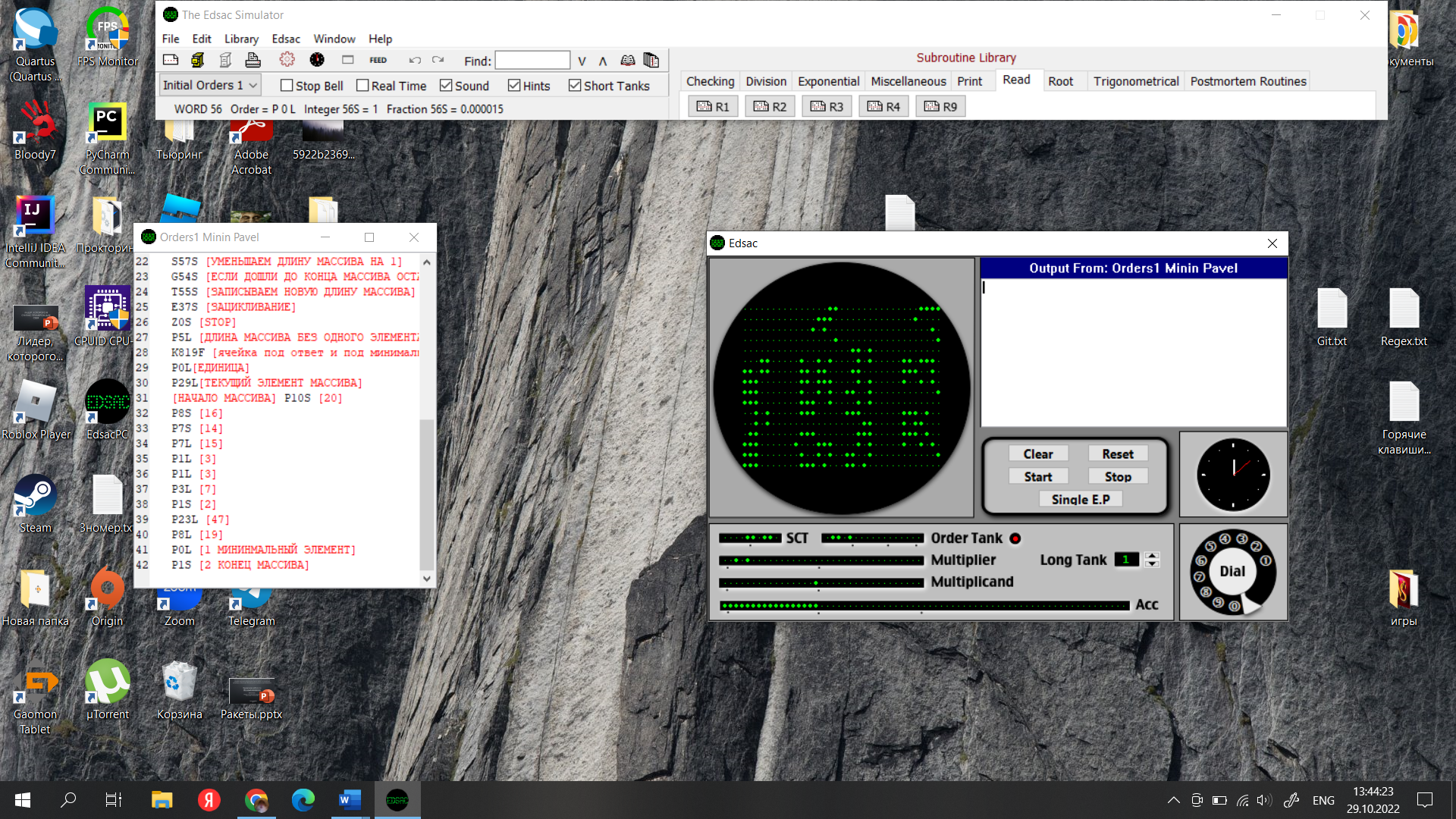
Код программы



Результат работы программы

Возьмем массив [20, 16, 14, 15, 3, 3, 7, 2, 47, 19, 1, 2]. Ожидаемый результат:1





Получившийся результат совпал с ожидаемым.

**Программа Initial Orders 2**

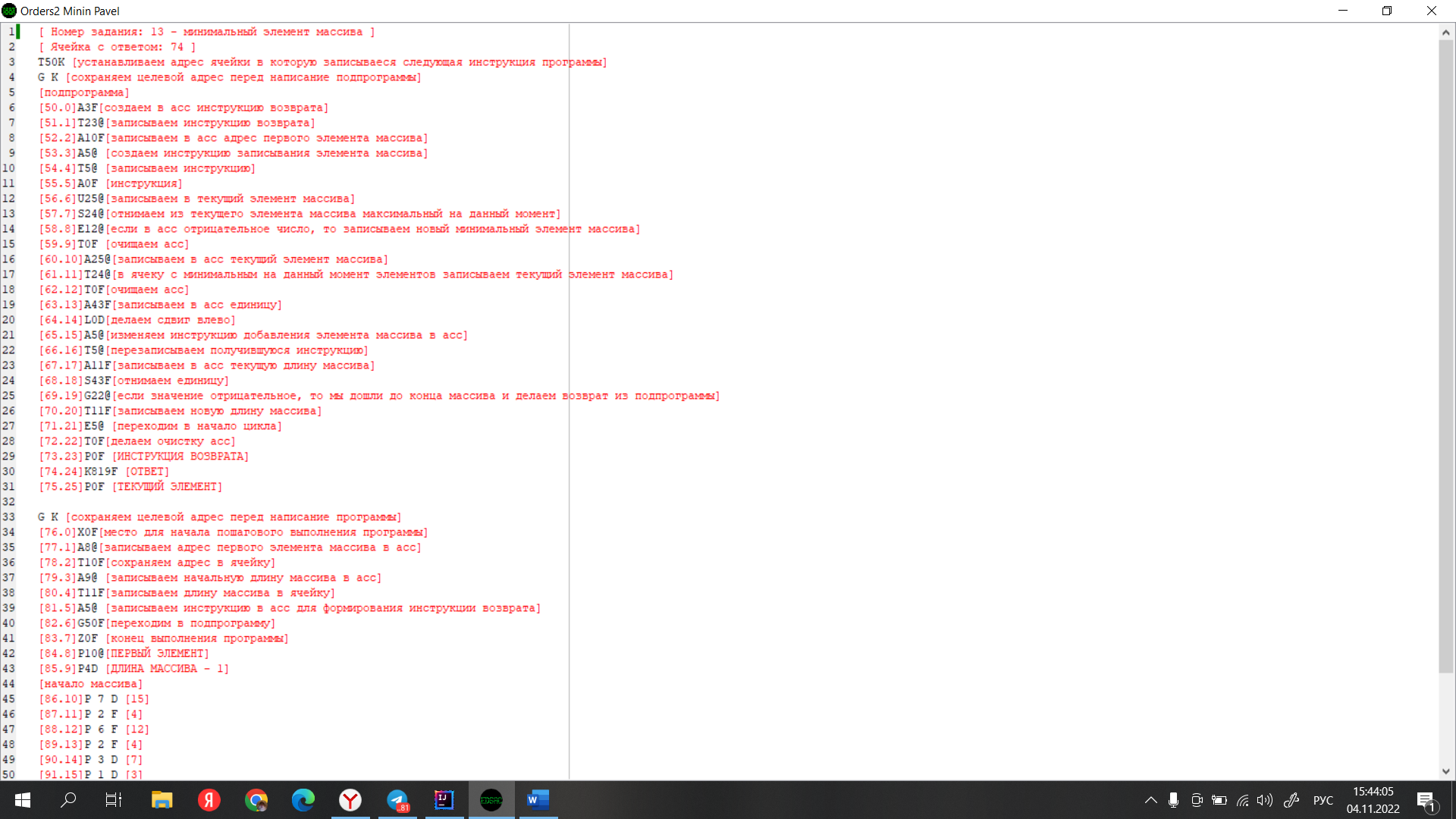
Программа, написанная для Initial Orders 2, делится на две части: подпрограмма и программа ее вызывающая. В Initial Orders 2 используется относительная адресация, что означает что адреса ячеек отсчитываются не с первой ячейки программы, а с начального адреса подпрограммы. Именно поэтому несмотря на то, что ячейка с ответом находится на 27 строке (не учитывая строки в которых написаны только комментарии) ее положение будет в 74 ячейке (так как заданный адрес начала подпрограммы в моем случае 50).

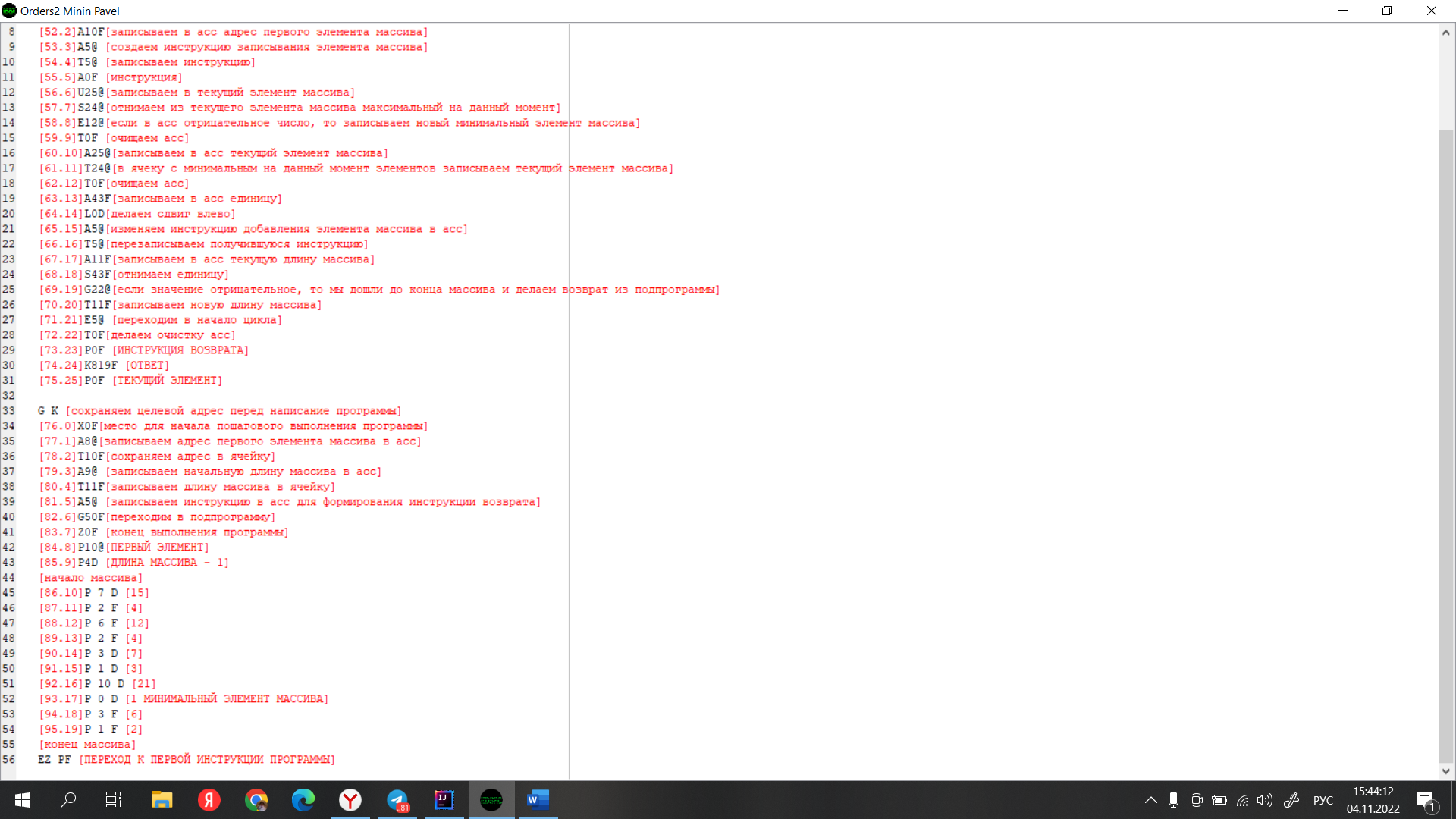
Логика программы остается такой же, как и в программе для Initials Orders 1.

Изменения:

1. D вместо L и F вместо S
2. Теперь можно обращаться к ячейкам по локальным адресам подпрограммы в которой они находятся, для чего используется код длины слова “@”.
3. Цикл прохождения по массиву и некоторые нужные значения вынесены в отдельную подпрограмму.
4. Теперь хранить свою единицу не нужно, так как константная единица хранится в 43 ячейке.

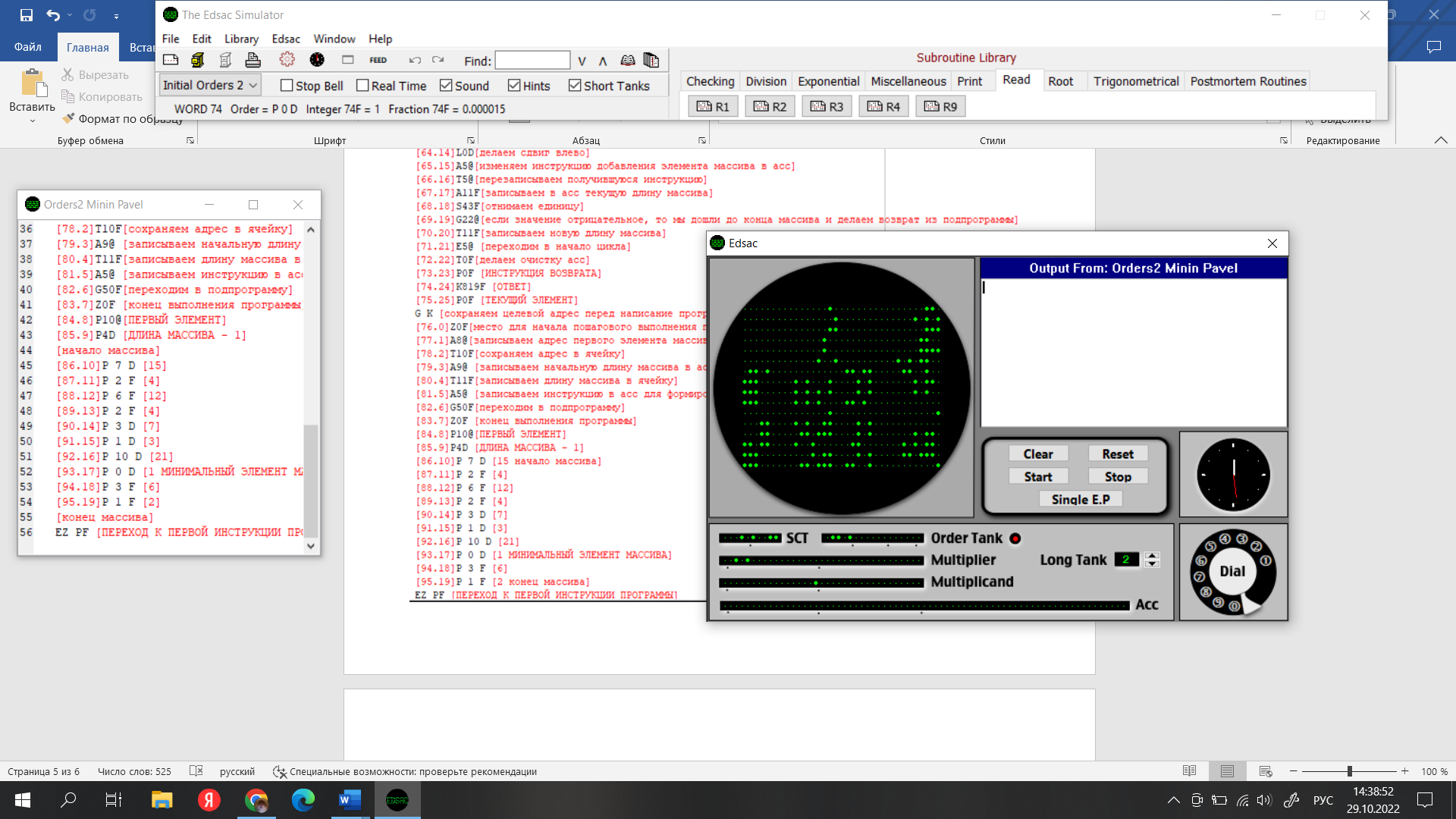
Код программы

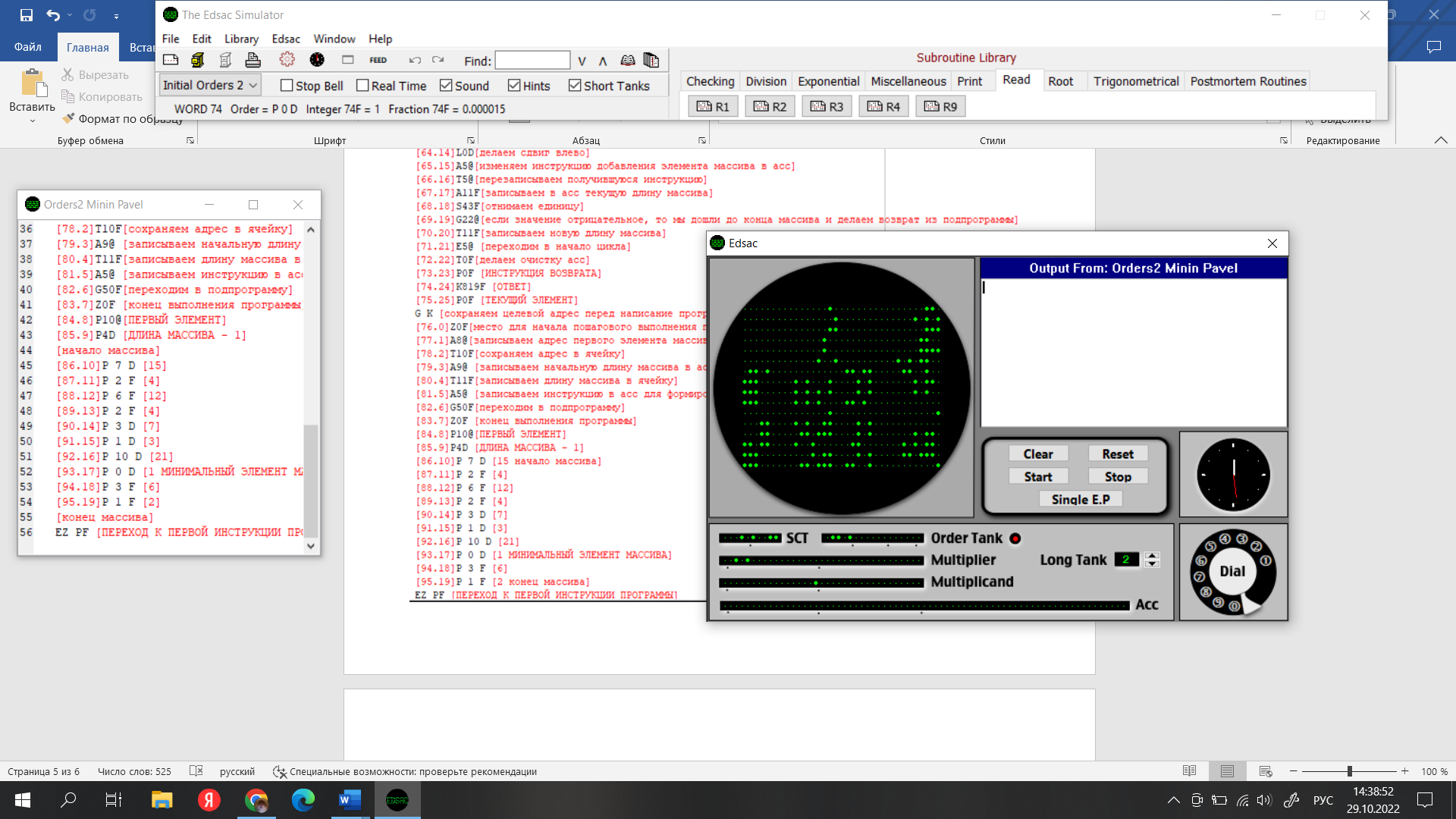




Результат работы программы

Рассмотрим массив [15, 4, 12, 4, 7, 3, 21, 1, 6, 2]. Ожидаемый результат:1





Результат работы программы совпал с ожидаемым.

**Вывод**

В ходе выполнения лабораторной работы были разработаны две программы в EDSAC для загрузчиков Initial Orders 1 и Initial Orders 2.

Обе программы выполняли поставленную задачу: нахождение минимального элемента массива. Как видно из результатов работ обоих программ для рассмотренных массивов получившийся ответ совпал с ожидаемым.