Санкт-Петербургский политехнический университет Петра Великого

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе №2**

**Дисциплина**: Низкоуровневое программирование

**Тема**: Программирование Edsac

Выполнил студент гр. 3530901/90003 А.А. Ундольский

(подпись)

Преподаватель А. О. Алексюк

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

Оглавление

[1. Техническое задание 3](#_Toc67442640)

[2. Метод решения 3](#_Toc67442641)

[3. Программа Initial Orders 1 3](#_Toc67442642)

[4. Работа программы Программа Initial Orders 1 8](#_Toc67442643)

[5. Программа Initial Orders 1 9](#_Toc67442644)

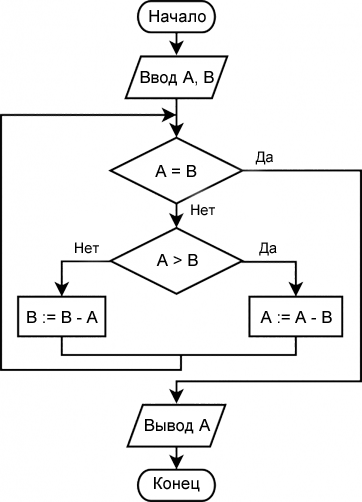
[6. Работа программы Программа Initial Orders 2 16](#_Toc67442645)

[7. Работа Адреса и правила кодирования исходных данных и результатов. Руководство программиста 17](#_Toc67442646)

# Техническое задание

Нахождение НОД массива чисел.

# Метод решения

Нахождение НОД осуществляется по алгоритму Евклида   
(методом вычитания):

Если A=B, то НОД найден и он равен одному из чисел, иначе необходимо большее из двух чисел заменить разностью его и меньшего.

# Программа Initial Orders 1

[Техническое задание - Определение НОД всех элементов массива]

[Используется Initial Orders 1]

[Результат (НОД всех элементов массива) записывается в ячейку 0]

[Длина массива задаётся в ячейке 98]

[Адрес первой ячейки массива задается в ячейке 99]

[Массив хранится в ячейках 100-107]

[31] T 108 S [Адрес ячейки, следующей за последней инструкцией программы]

[32] Z 0 S [Останов для отладки]

[33] T 0 S [Обнуление аккумулятора]

[В ячейке 1 хранится количество необработанных пар чисел (счетчик цикла)]

[Инициализация счетчика цикла]

[34] A 98 [<len>] S [загрузка в аккумулятор длины массива]

[35] S 97 [<c1>] S [вычитаем единицу, т.к. количество пар чисел на 1 меньше, чем длина массива]

[36] T 1 S [запись значения в ячейку 1]

[Инициализация адресных полей изменяемых инструкций]

[37] A 99 [<addr>] S [Загрузка в аккумулятор адреса 0-го элемента массива]

[38] L 0 L [Сдвиг аккумулятора на 1 разряд влево]

[39] A 61 [<r1>] S [Прибавляем код инструкции с нулевым полем адреса] [r1: A 0 S]

[40] U 61 [<r1>] S [Запись сформированной инструкции без обнуления аккумулятора]

[41] T 94 [<r1>] S [Запись сформированной инструкции с обнулением аккумулятора]

[42] A 99 [<addr>] S [Загрузка в аккумулятор адреса 0-го элемента массива]

[43] L 0 L [Сдвиг аккумулятора на 1 разряд влево]

[44] A 62 [<r2>] S [Прибавляем код инструкции с полем адреса 1] [r2: S 1 S]

[45] U 62 [<r2>] S [Запись сформированной инструкции без обнуления аккумулятора]

[46] T 69 [<r2>] S [Запись сформированной инструкции с обнулением аккумулятора]

[47] A 99 [<addr>] S [Загрузка в аккумулятор адреса 0-го элемента массива]

[48] L 0 L [Сдвиг аккумулятора на 1 разряд влево]

[49] A 64 [<w1>] S [Прибавляем код инструкции с нулевым полем адреса] [w1: U 0 S]

[50] T 64 [<w1>] S [Запись сформированной инструкции с обнулением аккумулятора]

[51] A 99 [<addr>] S [Загрузка в аккумулятор адреса 0-го элемента массива]

[52] L 0 L [Сдвиг аккумулятора на 1 разряд влево]

[53] A 68 [<w2>] S [Прибавляем код инструкции с полем адреса 1] [w2: T 1 S]

[54] U 68 [<w2>] S [Запись сформированной инструкции без обнуления аккумулятора]

[55] T 70 [<w2>] S [Запись сформированной инструкции с обнулением аккумулятора]

[Рабочий цикл]

[loop:]

[56] A 1 S [Загружаем в аккумулятор счетчик необработанных пар]

[57] S 97 [<c1>] S [Уменьшаем на 1]

[58] G 93 [<end>] S [Если результат меньше 0, идем на останов]

[59] T 1 S [Обновляем значение счетчика в ячейке]

[Поиск НОД для пары чисел]

[iter:]

[60] T 0 S [обнуляем аккумулятор]

[61] A 0 S [r1:] [ добавляем к аккумулятору первое число ]

[62] S 1 S [r2:] [ вычитаем из аккумулятора второе число ]

[63] G 68 [<invert>] S [ если результат отрицательный (первое число меньше второго), то идем к <invert> ]

[64] U 0 S [w1:] [ иначе в ячейку с первым числом кладем разность чисел, сохраняя ее в аккумуляторе ]

[65] S 97 [<c1>] S [ вычитаем из разности единицу и результат кладем в аккумулятор ]

[66] E 60 [<iter>] S [ если результат >=0, значит разность была >0, т.е. числа разные –> идем на следующую итерацию ]

[67] G 72 [<exit>] S [ числа одинаковые, НОД найден – выходим из цикла поиска НОД для пары чисел ]

[invert:]

[68] T 1 S [w2:] [ в ячейку со вторым числом кладем отрицательную разность, обнуляя аккумулятор ]

[69] S 1 S [r2:] [ вычитаем из нуля разность – фактически инвертируем знак! ]

[70] T 1 S [w2:] [ в ячейку со вторым числом кладем положительную разность, обнуляя аккумулятор ]

[71] E 60 [<iter>] S [ аккумулятор=0 –> идем к следующей итерации поиска НОД]

[exit:]

[Модификация адресных полей изменяемых инструкций для следующего шага рабочего цикла]

[72] T 0 S [обнуляем аккумулятор]

[73] A 97 [<c1>] S [загрузка в аккумулятор константы 1]

[74] L 0 L [сдвиг на 1 разряд влево]

[75] A 61 [<r1>] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[76] U 61 [<r1>] S [записываем сформированную инструкцию в память, без обнуления аккумулятора]

[77] T 94 [<r1>] S [записываем сформированную инструкцию в память]

[78] A 97 [<c1>] S [загрузка в аккумулятор константы 1]

[79] L 0 L [сдвиг на 1 разряд влево]

[80] A 62 [<r2>] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[81] U 62 [<r2>] S [записываем сформированную инструкцию в память, без обнуления аккумулятора]

[82] T 69 [<r2>] S [записываем сформированную инструкцию в память]

[83] A 97 [<c1>] S [загрузка в аккумулятор константы 1]

[84] L 0 L [сдвиг на 1 разряд влево]

[85] A 64 [<w1>] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[86] T 64 [<w1>] S [записываем сформированную инструкцию в память]

[87] A 97 [<c1>] S [загрузка в аккумулятор константы 1]

[88] L 0 L [сдвиг на 1 разряд влево]

[89] A 68 [<w2>] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[90] U 68 [<w2>] S [записываем сформированную инструкцию в память, без обнуления аккумулятора]

[91] T 70 [<w2>] S [записываем сформированную инструкцию в память, обнуляя аккумулятор]

[92] E 56 [<loop>] S [аккумулятор=0 -> повторяем рабочий цикл]

[end:]

[93] T 0 S [Обнуляем аккумулятор]

[94] A 0 S [r1:] [Загружаем в аккумулятор найденный НОД массива]

[95] T 0 S [Записываем его в ячейку 0]

[96] Z 0 S [останов ]

[c1:]

[97] P 0 L [константа 1 (0...0|0|0...0|1)]

[len:]

[98] P 4 S [длина массива (=8)]

[addr:]

[99] P 50 S [адрес первой ячейки массива (=100)]

[array:]

[100] P 4 L [9]

[101] P 9 S [18]

[102] P 15 S [30]

[103] P 30 S [60]

[104] P 60 S [120]

[105] P 18 S [36]

[106] P 16 L [33]

[107] P 12 S [24]

# Работа программы Программа Initial Orders 1

Исходные данные

len:



addr:



массив:









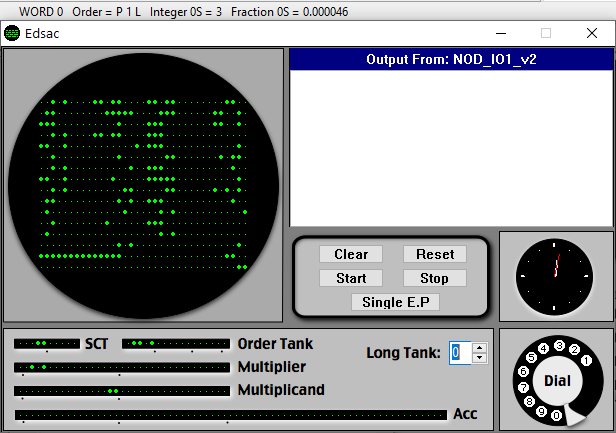








Результат работы:



# Программа Initial Orders 2

[Техническое задание - Определение НОД всех элементов массива]

[Используется Initial Orders 2]

[Длина массива задается в ячейке 55]

[Адрес первой ячейки массива задается в ячейке 56]

[Массив хранится в ячейках 57-64]

[Результат (НОД всех элементов массива) записывается в ячейку 0]

[Входные данные]

T 55 K [Входные данные размещаются с ячейки 55]

GK [@ = 55]

[@+0] P 4 F [<len> (=8) - длина массива]

[@+1] P 28 D [<addr> (=57) - адрес первой ячейки массива]

[@+2] P 4 D [9]

[@+3] P 9 F [18]

[@+4] P 15 F [30]

[@+5] P 30 F [60]

[@+6] P 60 F [120]

[@+7] P 18 F [36]

[@+8] P 16 D [33]

[@+9] P 12 F [24]

[Подпрограмма нахождения НОД массива]

[Входные параметры: адрес ячейки с длиной массива - 55, адрес первой ячейки массива - 56]

[Результат (НОД) - в ячейке 0]

T 200 K [Подпрограмма размещается с ячейки 200]

GK [@ = 200]

[@+0] A 3 F [формирование инструкции возврата в аккумуляторе]

[@+1] T 64 @ [<return>] [запись инструкции возврата, обнуление аккумулятора]

[В ячейке 1 хранится количество необработанных пар чисел (счетчик цикла)]

[Инициализация счетчика цикла]

[@+2] A 55 F [<len>] [загрузка в аккумулятор длины массива]

[@+3] S 65 @ [<c1>] [вычитаем единицу, т.к. количество пар чисел на 1 меньше, чем длина массива]

[@+4] T 1 F [запись значения в ячейку 1]

[Инициализация адресных полей изменяемых инструкций]

[@+5] A 56 F [<addr>] [Загрузка в аккумулятор адреса 0-го элемента массива]

[@+6] L 0 D [Сдвиг аккумулятора на 1 разряд влево]

[@+7] A 29 @ [<r1>] [Прибавляем код инструкции с нулевым полем адреса] [r1: A 0 F]

[@+8] U 29 @ [<r1>] [Запись сформированной инструкции без обнуления аккумулятора]

[@+9] T 62 @ [<r1>] [Запись сформированной инструкции с обнулением аккумулятора]

[@+10] A 56 F [<addr>] [Загрузка в аккумулятор адреса 0-го элемента массива]

[@+11] L 0 D [Сдвиг аккумулятора на 1 разряд влево]

[@+12] A 30 @ [<r2>] [Прибавляем код инструкции с полем адреса 1] [r2: S 1 F]

[@+13] U 30 @ [<r2>] [Запись сформированной инструкции без обнуления аккумулятора]

[@+14] T 37 @ [<r2>] [Запись сформированной инструкции с обнулением аккумулятора]

[@+15] A 56 F [<addr>] [Загрузка в аккумулятор адреса 0-го элемента массива]

[@+16] L 0 D [Сдвиг аккумулятора на 1 разряд влево]

[@+17] A 32 @ [<w1>] [Прибавляем код инструкции с нулевым полем адреса] [w1: U 0 F]

[@+18] T 32 @ [<w1>] [Запись сформированной инструкции с обнулением аккумулятора]

[@+19] A 56 F [<addr>] [Загрузка в аккумулятор адреса 0-го элемента массива]

[@+20] L 0 D [Сдвиг аккумулятора на 1 разряд влево]

[@+21] A 36 @ [<w2>] [Прибавляем код инструкции с полем адреса 1] [w2: T 1 F]

[@+22] U 36 @ [<w2>] [Запись сформированной инструкции без обнуления аккумулятора]

[@+23] T 38 @ [<w2>] [Запись сформированной инструкции с обнулением аккумулятора]

[Рабочий цикл]

[loop:]

[@+24] A 1 F [Загружаем в аккумулятор счетчик необработанных пар]

[@+25] S 65 @ [<c1>] [Уменьшаем на 1]

[@+26] G 61 @ [<end>] [Если результат меньше 0, то идем на выход из подпрограммы]

[@+27] T 1 F [Обновляем значение счетчика в ячейке]

[Поиск НОД для пары]

[@+28] T 0 F [обнуление аккумулятора]

[@+29] A 0 F [r1:] [добавляем к аккумулятору первое число]

[@+30] S 1 F [r2:] [вычитаем из аккумулятора второе число]

[@+31] G 36 @ [если результат отрицательный (первое число меньше второго), то идем к <invert>]

[@+32] U 0 F [w1:] [иначе в ячейку с первым числом кладем разность чисел, сохраняя ее в аккумуляторе]

[@+33] S 65 @ [вычитаем из разности единицу и результат кладем в аккумулятор]

[@+34] E 28 @ [если результат >=0, значит разность была >0, т.е. числа разные –> идем на следующую итерацию]

[@+35] G 40 @ [<exit>] [числа одинаковые, НОД найден – выходим из цикла поиска НОД для пары чисел]

[invert:]

[@+36] T 1 F [w2:] [в ячейку со вторым числом кладем отрицательную разность, обнуляя аккумулятор]

[@+37] S 1 F [r2:] [вычитаем из нуля разность – фактически инвертируем знак!]

[@+38] T 1 F [w2:] [в ячейку со вторым числом кладем положительную разность, обнуляя аккумулятор]

[@+39] E 28 @ [аккумулятор=0 –> идем к следующей итерации поиска НОД]

[exit:]

[Модификация адресных полей изменяемых инструкций для следующего шага рабочего цикла]

[@+40] T 0 F [обнуляем аккумулятор]

[@+41] A 65 @ [<c1>] [загрузка в аккумулятор константы 1]

[@+42] L 0 D [сдвиг на 1 разряд влево]

[@+43] A 29 @ [<r1>] [прибавляем код инструкции, исполненной на предыдущем шаге]

[@+44] U 29 @ [<r1>] [записываем сформированную инструкцию в память, без обнуления аккумулятора]

[@+45] T 62 @ [<r1>] [записываем сформированную инструкцию в память, с обнулением аккумулятора]

[@+46] A 65 @ [<c1>] [загрузка в аккумулятор константы 1]

[@+47] L 0 D [сдвиг на 1 разряд влево]

[@+48] A 30 @ [<r2>] [прибавляем код инструкции, исполненной на предыдущем шаге]

[@+49] U 30 @ [<r2>] [записываем сформированную инструкцию в память, без обнуления аккумулятора]

[@+50] T 37 @ [<r2>] [записываем сформированную инструкцию в память, с обнулением аккумулятора]

[@+51] A 65 @ [<c1>] [загрузка в аккумулятор константы 1]

[@+52] L 0 D [сдвиг на 1 разряд влево]

[@+53] A 32 @ [<w1>] [прибавляем код инструкции, исполненной на предыдущем шаге]

[@+54] T 32 @ [<w1>] [записываем сформированную инструкцию в память, с обнулением аккумулятора]

[@+55] A 65 @ [<c1>] [загрузка в аккумулятор константы 1]

[@+56] L 0 D [сдвиг на 1 разряд влево]

[@+57] A 36 @ [<r2>] [прибавляем код инструкции, исполненной на предыдущем шаге]

[@+58] U 36 @ [<r2>] [записываем сформированную инструкцию в память, без обнуления аккумулятора]

[@+59] T 38 @ [<r2>] [записываем сформированную инструкцию в память, с обнулением аккумулятора]

[@+60] E 24 @ [<loop>] [аккумулятор=0 -> повторяем рабочий цикл]

[end:]

[@+61] T 0 F [Обнуляем аккумулятор]

[@+62] A 0 F [r1:] [Загружаем в аккумулятор найденный НОД массива]

[@+63] T 0 F [Записываем его в ячейку 0]

[return:]

[@+64] E 0 F [Возврат из подпрограммы]

[c1:]

[@+65] P 0 D [константа 1]

[Вызывающая программа]

T 100 K [Программа размещается с ячейки 100]

GK [@ = 100]

[@+0] Z 0 F [останов для отладки]

[@+1] T 0 F [обнуление аккумулятора]

[@+2] A 2 @ [запись в аккумулятор текущего адреса для формирования в подпрограмме инструкции возврата]

[@+3] G 200 F [вызов подпрограммы нахождения НОД массива]

[@+4] Z 0 F [останов]

[@+5] EZPF [переход к первой инструкции программы]

# Работа программы Программа Initial Orders 2

Входные данные

len:



addr:



Массив:









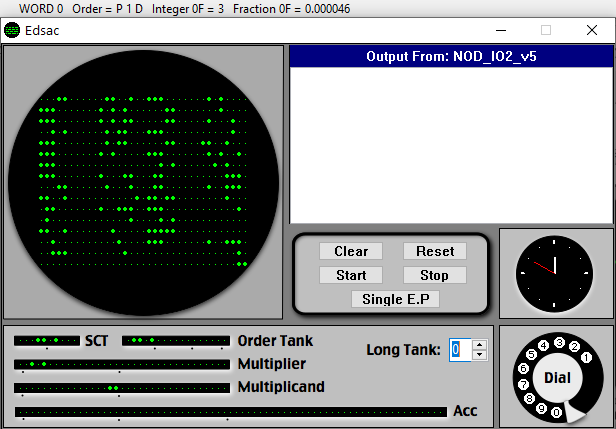








Результат работы:



# Работа Адреса и правила кодирования исходных данных и результатов. Руководство программиста

* 1. Загрузчик Initial Orders 1

Чтобы воспользоваться программой, необходимо:

* Записать исследуемый массив, начиная с 100 строки.
* В 98 строку записать длину массива.
* В строку 99 записать адрес первой ячейки массива.

Результат выполнения программы будет записан в ячейке 0.

* 1. Загрузчик InitialOrders 2

Чтобы воспользоваться программой, необходимо:

* Записать исследуемый массив, начиная с 13 строки

[@+2] P 4 D [9]

[@+3] P 9 F [18]

…)

Таким образом массив будет лежать с 57 ячейки, так как заданы относительные адреса – [@+2] P 4 D [9]; @=55 => 55+2=57

* В 11 строку записать длину массива

[@+0] P 4 F [<len> (=8) - длина массива]

То есть длина массива лежит в 55 ячейке.

Аналогично тому, что выше – @=55 => 55+0=55

* В строку 12 записать адрес первой ячейки массива.

[@+1] P 28 D [<addr> (=57) - адрес первой ячейки массива]

Значение адрес первой ячейки массива будет находится в ячейке 56,   
так как @=55 => 55+1=56.

Результат выполнения программы будет записан в ячейке 0.