МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И

КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

«Основы профессиональной деятельности»

Выполнение циклических программ

Вариант №1302

***Выполнил:***

Студент группы P3107

Шишкин Артём Владимирович

***Принял:***

Святослав Осипов

Санкт-Петербург, 2025

**Содержание**

[**Задание** 3](#_Toc191471721)

[**Текст исходной программы** 3](#_Toc191471722)

[**Описание программы** 5](#_Toc191471723)

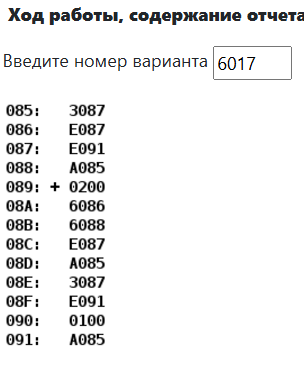
[**Расположение данных в памяти** 5](#_Toc191471724)

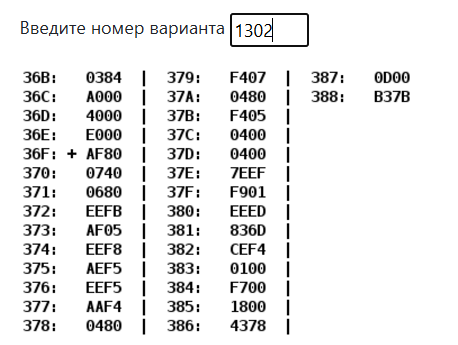
[**Таблица трассировки** 5](#_Toc191471725)

[**Заключение** 8](#_Toc191471726)

2

# **Задание**





# **Текст исходной программы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес** | **Код**  **Команды** | **Мнемоника** | **Комментарии** |  |
|  |  | ORG 0x36B |  |  |
| 36B | 0384 | first: WORD $arr0 | Первый элемент массива |  |
| 36C | A000 | next: WORD 0xA000 | Следующий элемент |  |
| 36D | 4000 | length: WORD 0x4000 | Длина массива |  |
| 36E | E000 | result: WORD 0xE000 | Результат |  |
| 36F | AF80 | START: LD #0x80 | Прямая загрузка FF80 -> AC | Кладем в результат максимальное положительное число |
| 370 | 0740 | DEC | AC – 1 -> AC |
| 371 | 0680 | SWAB | Обмен ст. и мл. байта |
| 372 | EEFB | ST result | Прямое относительное сохранение AC -> M(36E) |
| 373 | AF05 | LD #5 | Прямая загрузка 0005 -> AC | Устанавливаем длину массива |
| 374 | EEF8 | ST length | Прямое относительное сохранение AC -> M(36D) |
| 375 | AEF5 | LD first | Прямая относительная загрузка M -> AC (36B) | Устанавливаем первый элемент |
| 376 | EEF5 | ST next | Прямое относительное сохранение AC -> M(36C) |
| 377 | AAF4 | repeat: LD (next)+ | Косвенная автоинкрементная загрузка M -> AC (36C); Зн(36C) += 1 | Начало цикла |
| 378 | 0480 | ROR | AC и C сдвигается вправо | Проверяем делимость на 2 |
| 379 | F407 | BHIS x | Если C == 1, то IP = IP + 1 + 7 -> IP (381) |
| 37A | 0480 | ROR | AC и C сдвигается вправо | Проверяем делимость на 4 |
| 37B | F405 | BHIS x | Если C == 1, то IP = IP + 1 + 7 -> IP (381) |
| 37C | 0400 | ROL | AC и C сдвигается влево | Возвращаем исходное число |
| 37D | 0400 | ROL | AC и C сдвигается влево |
| 37E | 7EEF | CMP result | Установить флаги по результату AC – M(36E) | Если число меньше результата, то оно новый результат |
| 37F | F901 | BGE x | Если N==V, то IP = IP + 1 + 1 -> IP |
| 380 | EEED | ST result | Прямое относительное сохранение AC -> M(36E) |
| 381 | 836D | x: LOOP $length | Зн(36D) – 1 -> Яч(36D); Если Зн(36D) <= 0, то IP + 1 -> IP |
| 382 | CEF4 | JUMP repeat | Прямой относительный прыжок IP – 12 + 1 -> IP (377) | Конец цикла |
| 383 | 0100 | HLT |  |  |
| 384 | F700 | arr0: WORD 0xF700 | Массив |  |
| 385 | 1800 | arr1: WORD  0x1800 |  |
| 386 | 4378 | arr2: WORD 0x4378 |  |
| 387 | 0D00 | arr3: WORD  0x0D00 |  |
| 388 | D37B | arr4: WORD  0xD37B |  |

# **Описание программы**

Программа находит минимальный элемент массива, делящийся на 4

ОДЗ:

* result ϵ [-215, -215+4, ... , 215–4]
* length ϵ [1; 27 – 1]
* first ϵ [0 ; 36B – length] ∨ [389; 7FF – length + 1]
* next ϵ [first; first + length]
* Элементы массива arr[i] ϵ [-215; 215–1]

# **Расположение данных в памяти**

* 36B, 36C, 36D, 36E, 384, 385, 386, 387, 378 – исходные данные;
* 36E – результат.
* 36F - 383 – команды.

36F – первая исполняемая команда

383 – последняя исполняемая команда

# **Таблица трассировки**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров процессора после выполнения команды | | | | | | | | Ячейка, содержимое которой изменилось после выполнения команды | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 36F | AF80 | 370 | AF80 | 36F | FF80 | 000 | FF80 | FF80 | 1000 |  |  |
| 370 | 0740 | 371 | 0740 | 370 | 0740 | 000 | 0370 | FF7F | 1001 |  |  |
| 371 | 0680 | 372 | 0680 | 371 | 0680 | 000 | 0371 | 7FFF | 0001 |  |  |
| 372 | EEFB | 373 | EEFB | 36E | 7FFF | 000 | FFFB | 7FFF | 0001 | 36E | 7FFF |
| 373 | AF05 | 374 | AF05 | 373 | 0005 | 000 | 0005 | 0005 | 0001 |  |  |
| 374 | EEF8 | 375 | EEF8 | 36D | 0005 | 000 | FFF8 | 0005 | 0001 | 36D | 0005 |
| 375 | AEF5 | 376 | AEF5 | 36B | 0384 | 000 | FFF5 | 0384 | 0001 |  |  |
| 376 | EEF5 | 377 | EEF5 | 36C | 0384 | 000 | FFF5 | 0384 | 0001 | 36C | 0384 |
| 377 | AAF4 | 378 | AAF4 | 384 | F700 | 000 | FFF4 | F700 | 1001 | 36C | 0385 |
| 378 | 0480 | 379 | 0480 | 378 | 0480 | 000 | 0378 | FB80 | 1010 |  |  |
| 379 | F407 | 37A | F407 | 379 | F407 | 000 | 0379 | FB80 | 1010 |  |  |
| 37A | 0480 | 37B | 0480 | 37A | 0480 | 000 | 037A | 7DC0 | 0000 |  |  |
| 37B | F405 | 37C | F405 | 37B | F405 | 000 | 037B | 7DC0 | 0000 |  |  |
| 37C | 0400 | 37D | 0400 | 37C | 0400 | 000 | 037C | FB80 | 1010 |  |  |
| 37D | 0400 | 37E | 0400 | 37D | 0400 | 000 | 037D | F700 | 1001 |  |  |
| 37E | 7EEF | 37F | 7EEF | 36E | 7FFF | 000 | FFEF | F700 | 0011 |  |  |
| 37F | F901 | 380 | F901 | 37F | F901 | 000 | 037F | F700 | 0011 |  |  |
| 380 | EEED | 381 | EEED | 36E | F700 | 000 | FFED | F700 | 0011 | 36E | F700 |
| 381 | 836D | 382 | 836D | 36D | 0004 | 000 | 0003 | F700 | 0011 | 36D | 0004 |
| 382 | CEF4 | 377 | CEF4 | 382 | 0377 | 000 | FFF4 | F700 | 0011 |  |  |
| 377 | AAF4 | 378 | AAF4 | 385 | 1800 | 000 | FFF4 | 1800 | 0001 | 36C | 0386 |
| 378 | 0480 | 379 | 0480 | 378 | 0480 | 000 | 0378 | 8C00 | 1010 |  |  |
| 379 | F407 | 37A | F407 | 379 | F407 | 000 | 0379 | 8C00 | 1010 |  |  |
| 37A | 0480 | 37B | 0480 | 37A | 0480 | 000 | 037A | 4600 | 0000 |  |  |
| 37B | F405 | 37C | F405 | 37B | F405 | 000 | 037B | 4600 | 0000 |  |  |
| 37C | 0400 | 37D | 0400 | 37C | 0400 | 000 | 037C | 8C00 | 1010 |  |  |
| 37D | 0400 | 37E | 0400 | 37D | 0400 | 000 | 037D | 1800 | 0011 |  |  |
| 37E | 7EEF | 37F | 7EEF | 36E | F700 | 000 | FFEF | 1800 | 0000 |  |  |
| 37F | F901 | 381 | F901 | 37F | F901 | 000 | 0001 | 1800 | 0000 |  |  |
| 381 | 836D | 382 | 836D | 36D | 0003 | 000 | 0002 | 1800 | 0000 | 36D | 0003 |
| 382 | CEF4 | 377 | CEF4 | 382 | 0377 | 000 | FFF4 | 1800 | 0000 |  |  |
| 377 | AAF4 | 378 | AAF4 | 386 | 4378 | 000 | FFF4 | 4378 | 0000 | 36C | 0387 |
| 378 | 0480 | 379 | 0480 | 378 | 0480 | 000 | 0378 | 21BC | 0000 |  |  |
| 379 | F407 | 37A | F407 | 379 | F407 | 000 | 0379 | 21BC | 0000 |  |  |
| 37A | 0480 | 37B | 0480 | 37A | 0480 | 000 | 037A | 10DE | 0000 |  |  |
| 37B | F405 | 37C | F405 | 37B | F405 | 000 | 037B | 10DE | 0000 |  |  |
| 37C | 0400 | 37D | 0400 | 37C | 0400 | 000 | 037C | 21BC | 0000 |  |  |
| 37D | 0400 | 37E | 0400 | 37D | 0400 | 000 | 037D | 4378 | 0000 |  |  |
| 37E | 7EEF | 37F | 7EEF | 36E | F700 | 000 | FFEF | 4378 | 0000 |  |  |
| 37F | F901 | 381 | F901 | 37F | F901 | 000 | 0001 | 4378 | 0000 |  |  |
| 381 | 836D | 382 | 836D | 36D | 0002 | 000 | 0001 | 4378 | 0000 | 36D | 0002 |
| 382 | CEF4 | 377 | CEF4 | 382 | 0377 | 000 | FFF4 | 4378 | 0000 |  |  |
| 377 | AAF4 | 378 | AAF4 | 387 | 0D00 | 000 | FFF4 | 0D00 | 0000 | 36C | 0388 |
| 378 | 0480 | 379 | 0480 | 378 | 0480 | 000 | 0378 | 0680 | 0000 |  |  |
| 379 | F407 | 37A | F407 | 379 | F407 | 000 | 0379 | 0680 | 0000 |  |  |
| 37A | 0480 | 37B | 0480 | 37A | 0480 | 000 | 037A | 0340 | 0000 |  |  |
| 37B | F405 | 37C | F405 | 37B | F405 | 000 | 037B | 0340 | 0000 |  |  |
| 37C | 0400 | 37D | 0400 | 37C | 0400 | 000 | 037C | 0680 | 0000 |  |  |
| 37D | 0400 | 37E | 0400 | 37D | 0400 | 000 | 037D | 0D00 | 0000 |  |  |
| 37E | 7EEF | 37F | 7EEF | 36E | F700 | 000 | FFEF | 0D00 | 0000 |  |  |
| 37F | F901 | 381 | F901 | 37F | F901 | 000 | 0001 | 0D00 | 0000 |  |  |
| 381 | 836D | 382 | 836D | 36D | 0001 | 000 | 0000 | 0D00 | 0000 | 36D | 0001 |
| 382 | CEF4 | 377 | CEF4 | 382 | 0377 | 000 | FFF4 | 0D00 | 0000 |  |  |
| 377 | AAF4 | 378 | AAF4 | 388 | D37B | 000 | FFF4 | D37B | 1000 | 36C | 0389 |
| 378 | 0480 | 379 | 0480 | 378 | 0480 | 000 | 0378 | 69BD | 0011 |  |  |
| 379 | F407 | 381 | F407 | 379 | F407 | 000 | 0007 | 69BD | 0011 |  |  |
| 381 | 836D | 383 | 836D | 36D | 0000 | 000 | FFFF | 69BD | 0011 | 36D | 0000 |

# **Ассемблер**

ORG 0x36B

first: WORD $arr0

next: WORD 0xA000

length: WORD 0x4000

result: WORD 0xE000

START:

LD #0x80

DEC

SWAB

ST result

LD #0x05

ST length

LD first

ST next

repeat: LD (next)+

ROR

BHIS c\_is\_1

ROR

BHIS c\_is\_1

ROL

ROL

CMP result

BGE c\_is\_1

ST result

c\_is\_1: LOOP $length

JUMP repeat

HLT

arr0: WORD 0xF700

arr1: WORD 0x1800

arr2: WORD 0x4378

arr3: WORD 0x0D00

arr4: WORD 0xD37B

# **Доп**

**JUMP ((L)+)**

temp: WORD 0x2

ORG 0x15

START: PUSH

PUSHF

LD (L)+

ST temp

POPF

POP

JUMP (temp)

HLT

ORG 0x55

L: WORD 0x111

ORG 0x111

WORD 0x222

# **Заключение**

Во время выполнения лабораторной работы я научился работать в БЭВМ с массивами, ветвлением и циклами. Я изучил режимы адресации и цикл выполнения таких команд, как LOOP и JUMP.

org 0x10

huge: word 0xffff

a: cmp huge

bge (b)

b: word 0x133

start: jump a

cla

cla

cla

cla

hlt

end