Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

По Основам Профессиональной Деятельности

Вариант 9504

Выполнил:

Ларионов Владислав Васильевич

Группа P3109

Практик:

Ткешелашвили Н. М.

Санкт-Петербург 2025

Содержание

[**Задание** 3](#_Toc197455738)

[**Выполнение задания** 4](#_Toc197455739)

[1.1 Текст исходной программы 4](#_Toc197455740)

[1.2 Описание программы 8](#_Toc197455741)

[1.3 Получение слова 9](#_Toc197455742)

[1.4 Таблица трассировки 9](#_Toc197455743)

[1.5 Доп задание 10](#_Toc197455744)

[**Вывод:** 13](#_Toc197455745)

# **Задание**

1. Программа осуществляет асинхронный вывод данных на ВУ-3
2. Программа начинается с адреса 40D16. Размещаемая строка находится по адресу 55516.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Вывод строки начинается с вывода количества символов (1 байт), и должен быть завершен по выводу их необходимого количества.

# **Выполнение задания**

# 1.1 Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
| 40D | 063A | LENADR | Ячейка с адресом длины |
| 40E | 0000 | LEN | Длина слова для цикла |
| 40F | 063B | ADR | Ячейка для инкрементирования адреса результата |
| 410 | 0200 | CLA | Очистка аккумулятора 0->AC |
| 411 | 1205 | IN 7 | Чтение содержимого регистра SR ВУ-3 (ожидание ввода длины) |
| 412 | 2F40 | AND #40 | Проверка 6 бита SR=0 (если не готов) |
| 413 | F0FD | BEQ (IP-3) | Спин-луп |
| 414 | 1204 | IN 6 | Ввод длины слова |
| 415 | EEF8 | ST IP-8 | Сохранение длины во временную переменную |
| 416 | E8F6 | ST (IP-9) | Сохранение длины слова в LENGTH |
| 417 | 0200 | CLA | Очистка аккумулятора 0->AC |
| 418 | AEF5 | LD IP-11 | Загрузка длины LEN->AC |
| 419 | 0740 | DEC | Декремент AC-=1 |
| 41A | EEF3 | ST IP-13 | Сохранение полученного результата |
| 41B | F213 | BMI (IP+19) | Переход на 0x422 если N==1 |
| 41C | 0200 | CLA | Очистка аккумулятора 0->AC |
| 41D | 1205 | IN 7 | Чтение содержимого регистра SR ВУ-3 (ожидание ввода символа) |
| 41E | 2F40 | AND #40 | Проверка готовности, 6 бит SR=0 |
| 41F | F0FD | BEQ (IP-3) | Спин-луп, если не готов |
| 420 | 1204 | IN 6 | Ввод символа |
| 421 | 0680 | SWAB | Перемещение символа в старший разряд AC |
| 422 | E8EC | ST (IP-20) | Сохранение его в ячейке RESULT |
| 423 | AEEA | LD IP-22 | Загрузка длины LEN->AC |
| 424 | 0740 | DEC | Декремент AC-=1 |
| 425 | EEE8 | ST IP-24 | Сохранение полученного результата |
| 426 | F208 | BMI (IP+8) | Переход на 0x422 если N==1 |
| 427 | 0200 | CLA | Очистка аккумулятора 0->AC |
| 428 | 1205 | IN 7 | Чтение содержимого регистра SR ВУ-3 (ожидание ввода символа) |
| 429 | 2F40 | AND #40 | Проверка готовности, 6 бит SR=0 |
| 42A | F0FD | BEQ (IP-3) | Спин-луп, если не готов |
| 42B | A8E3 | LD (IP-29) | Загрузка предыдущего символа |
| 42C | 1204 | IN 6 | Чтение следующего символа в младший байт AC |
| 42D | EAE1 | ST (IP-31)+ | Сохранение по адресу из ячейки ADR с увлечением адреса в ней на 1 |
| 42E | CEE9 | JUMP (IP-23) | Переход к адресу 0x40A |
| 42F | 0200 | CLA | Очистка аккумулятора 0->AC |
| 430 | 0100 | HLT | Остановка программы |
| --------- | -------------- | --------------------- | ------------------------------------------------------------- |
| 554 | 0000 | LENGTH | Ячейка для хранения длины слова |
| 555 | 0000 | RESULT | Ячейка для хранения символов |

Таблица 2 – исходная программа

## 1.2 Описание программы

**Предназначение программы:**

Асинхронный ввод с ВУ-3, сначала вводится длина, потом эти символы (пока не будут введены все)

**Область представления программы:**

RESULT – 16-разрядное значение 2 символов, где старший байт – код первого символа, а младший - второго

LENGTH – 8-разрядное беззнаковое число, хранит длину слова

ADR – 11-разрядное беззнаковое число. Ячейка хранит адрес начала слова

LEN – 8-разрядное беззнаковое число ячейка для декремента длины слова

Изображение выглядит как текст, снимок экрана, меню, Симметрия

Контент, сгенерированный ИИ, может содержать ошибки.

**Область допустимых значений программы:**

LENGTH, LEN , значит, максимум можно ввести 255 символов

SUM – 8-разрядный код символа за исключением служебных символов

## 1.3 Получение слова

ВОРОН

**ISO-8859-5:**

В – B2 (1011 0010)

О – BE (1011 1110)

Р – C0 (1100 0000)

О – BE (1011 1110)

Н – BD (1011 1101)

**UNICODE:**

В – \u0412

О – \u041e

Р – \u0420

О – \u041e

Н – \u041d

## 1.4 Таблица трассировки

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Адр** | **Знчн** | **IP** | **CR** | **AR** | **DR** | **SP** | **BR** | **AC** | **NZVC** | **Адр** | **Знчн** |
| 410 | 0200 | 411 | 0200 | 410 | 0200 | 000 | 0410 | 0000 | 0100 |  |  |
| 411 | 1207 | 412 | 1207 | 411 | 1207 | 000 | 0411 | 0040 | 0100 |  |  |
| 412 | 2F40 | 413 | 2F40 | 412 | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 413 | F0FD | 414 | F0FD | 413 | F0FD | 000 | 0413 | 0040 | 0000 |  |  |
| 414 | 1206 | 415 | 1206 | 414 | 1206 | 000 | 0414 | 0002 | 0000 |  |  |
| 415 | E8F8 | 416 | E8F8 | 554 | 0002 | 000 | FFF8 | 0002 | 0000 | 554 | 0002 |
| 416 | A8F7 | 417 | A8F7 | 554 | 0002 | 000 | FFF7 | 0002 | 0000 |  |  |
| 417 | E8F5 | 418 | E8F5 | 553 | 0002 | 000 | FFF5 | 0002 | 0000 | 553 | 0002 |
| 418 | 0200 | 419 | 0200 | 418 | 0200 | 000 | 0418 | 0000 | 0100 |  |  |
| 419 | A8F4 | 41A | A8F4 | 554 | 0002 | 000 | FFF4 | 0002 | 0000 |  |  |
| 41A | 0740 | 41B | 0740 | 41A | 0740 | 000 | 041A | 0001 | 0001 |  |  |
| 41B | E8F2 | 41C | E8F2 | 554 | 0001 | 000 | FFF2 | 0001 | 0001 | 554 | 0001 |
| 41C | F213 | 41D | F213 | 41C | F213 | 000 | 041C | 0001 | 0001 |  |  |
| 41D | 0200 | 41E | 0200 | 41D | 0200 | 000 | 041D | 0000 | 0101 |  |  |
| 41E | 1207 | 41F | 1207 | 41E | 1207 | 000 | 041E | 0040 | 0101 |  |  |
| 41F | 2F40 | 420 | 2F40 | 41F | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 420 | F0FD | 421 | F0FD | 420 | F0FD | 000 | 0420 | 0040 | 0001 |  |  |
| 421 | 1206 | 422 | 1206 | 421 | 1206 | 000 | 0421 | 00B2 | 0001 |  |  |
| 422 | 0680 | 423 | 0680 | 422 | 0680 | 000 | 0422 | B200 | 1001 |  |  |
| 423 | E8EB | 424 | E8EB | 555 | B200 | 000 | FFEB | B200 | 1001 | 555 | B200 |
| 424 | A8E9 | 425 | A8E9 | 554 | 0001 | 000 | FFE9 | 0001 | 0001 |  |  |
| 425 | 0740 | 426 | 0740 | 425 | 0740 | 000 | 0425 | 0000 | 0101 |  |  |
| 426 | E8E7 | 427 | E8E7 | 554 | 0000 | 000 | FFE7 | 0000 | 0101 | 554 | 0000 |
| 427 | F208 | 428 | F208 | 427 | F208 | 000 | 0427 | 0000 | 0101 |  |  |
| 428 | 0200 | 429 | 0200 | 428 | 0200 | 000 | 0428 | 0000 | 0101 |  |  |
| 429 | 1207 | 42A | 1207 | 429 | 1207 | 000 | 0429 | 0040 | 0101 |  |  |
| 42A | 2F40 | 42B | 2F40 | 42A | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 42B | F0FD | 42C | F0FD | 42B | F0FD | 000 | 042B | 0040 | 0001 |  |  |
| 42C | A8E2 | 42D | A8E2 | 555 | B200 | 000 | FFE2 | B200 | 1001 |  |  |
| 42D | 1206 | 42E | 1206 | 42D | 1206 | 000 | 042D | B2BE | 1001 |  |  |
| 42E | EAE0 | 42F | EAE0 | 555 | B2BE | 000 | FFE0 | B2BE | 1001 | 555 | B2BE |
| 42F | CEE9 | 419 | CEE9 | 42F | 0419 | 000 | FFE9 | B2BE | 1001 |  |  |
| 419 | A8F4 | 41A | A8F4 | 554 | 0000 | 000 | FFF4 | 0000 | 0101 |  |  |
| 41A | 0740 | 41B | 0740 | 41A | 0740 | 000 | 041A | FFFF | 1000 |  |  |
| 41B | E8F2 | 41C | E8F2 | 554 | FFFF | 000 | FFF2 | FFFF | 1000 | 554 | FFFF |
| 41C | F213 | 430 | F213 | 41C | F213 | 000 | 0013 | FFFF | 1000 |  |  |
| 430 | A8DC | 431 | A8DC | 553 | 0002 | 000 | FFDC | 0002 | 0000 |  |  |
| 431 | E8DC | 432 | E8DC | 554 | 0002 | 000 | FFDC | 0002 | 0000 | 554 | 0002 |
| 432 | 0200 | 433 | 0200 | 432 | 0200 | 000 | 0432 | 0000 | 0100 |  |  |
| 433 | E8D9 | 434 | E8D9 | 553 | 0000 | 000 | FFD9 | 0000 | 0100 | 553 | 0000 |
| 434 | 0100 | 435 | 0100 | 434 | 0100 | 000 | 0434 | 0000 | 0100 |  |  |

Таблица 2 – трассировка

## 1.5 Доп задание

C ВУ-3 вводится знаковое 8-разрядное число. Программа выводит ВУ-7 (семисегментный индикатор) модуль введённого числа.

РЕШЕНИЕ:

ORG 0x10

START:

    CLA

INPUT:

    IN 7

    AND #0x40

    BEQ INPUT

    IN 6

    ST TEMP

    LD TEMP

    AND #0x80

    BEQ CONVERT

    LD TEMP

    NEG

    ST TEMP

CONVERT:

    CLA

    ST HUNDREDS

    ST TENS

    ST UNITS

COUNT\_HUNDREDS:

    LD TEMP

    CMP #100

    BMI COUNT\_TENS

    SUB #100

    ST TEMP

    LD HUNDREDS

    ADD #1

    ST HUNDREDS

    JUMP COUNT\_HUNDREDS

COUNT\_TENS:

    LD TEMP

    CMP #10

    BMI SAVE\_UNITS

    SUB #10

    ST TEMP

    LD TENS

    ADD #1

    ST TENS

    JUMP COUNT\_TENS

SAVE\_UNITS:

    LD TEMP

    ST UNITS

PRINT\_HUNDREDS:

    LD HUNDREDS

    BEQ ZERO\_HUNDREDS

    ADD #0x20

    OUT 0x14

    JUMP PRINT\_TENS

ZERO\_HUNDREDS:

    LD TENS

    BEQ PRINT\_UNITS

    ADD #0x10

    OUT 0x14

    JUMP PRINT\_UNITS

PRINT\_TENS:

    LD TENS

    ADD #0x10

    OUT 0x14

PRINT\_UNITS:

    LD UNITS

    OUT 0x14

    HLT

ORG 0x90

TEMP: WORD 0

HUNDREDS: WORD 0

TENS: WORD 0

UNITS: WORD 0

# **Вывод:**

В ходе выполнения данной лабораторной работы я научился работать с устройствами ввода-вывода, повторил навыки трассировки, научился кодировать символы и трассировать закодированные значения, подаваемые на устройство ввода.