

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет ИТМО»  
Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по основам  
профессиональной деятельности №4**

Вариант 15601

Выполнил: Снагин Станислав Максимович  
Проверил: Блохина Елена Николаевна  
Группа: Р3115

Санкт Петербург 2025 г.

## Содержание

Задания и цели работы.....	3
Ход работы.....	4
Текст исходной программы.....	4
Текст подпрограммы.....	6
Данные подпрограммы.....	6
Данные.....	7
Назначение программы, подпрограммы и комплекса программ и реализуемая ею функция.....	7
Область представления.....	8
Область допустимых значений.....	8
Расположение данных в памяти ЭВМ программы, исходных данных и результатов:.....	9
Программа.....	9
Подпрограмма.....	10
Адреса первой и последней выполняемой команды.....	10
В программе.....	10
В подпрограмме.....	10
Трассировка программы.....	10
Вывод.....	13

## Задания и цели работы

**Цель работы** — изучение способов связи между программными модулями, команды обращения к подпрограмме и исследование порядка функционирования БЭВМ при выполнении комплекса взаимосвязанных программ.

**Задание** — по выданному преподавателем варианту восстановить текст заданного варианта программы и подпрограммы (программного комплекса), определить их предназначение и составить описание, определить область представления и область допустимых значений для исходных данных и возвращаемых значений подпрограммы, выполнить трассировку программного комплекса.

39E: + 0200		3AC: 0800		3BA: YYYY		729: EC01
39F: EE1C		3AD: 0740		3BB: XXXX		72A: 0A00
3A0: AE19		3AE: 6E0D		3BC: FF3A		72B: 0F6C
3A1: 0700		3AF: EE0C		-----		72C: 00C1
3A2: 0C00		3B0: AE0A		71F: AC01		
3A3: D71F		3B1: 0700		720: F204		
3A4: 0800		3B2: 0C00		721: F003		
3A5: 0700		3B3: D71F		722: 7E08		
3A6: 6E15		3B4: 0800		723: F004		
3A7: EE14		3B5: 0740		724: F803		
3A8: AE10		3B6: 4E05		725: 4C01		
3A9: 0740		3B7: EE04		726: 6E05		
3AA: 0C00		3B8: 0100		727: CE01		
3AB: D71F		3B9: ZZZZ		728: AE02		

## Ход работы

### Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарии
39E+	0200	CLA	Начало программы
39F	EE1C	ST IP + 28	(AC=0) AC → D, отчистка результата
3A0	AE19	LD IP + 25	Y → AC
3A1	0700	INC	AC = Y + 1
3A2	0C00	PUSH	Передача Y + 1 (аргумента) в стек (0x7FF)
3A3	D71F	CALL 0x71F	Вызов подпрограммы F(Y + 1) (0x7FE)
3A4	0800	POP	Получаем результат подпрограммы в AC
3A5	0700	INC	AC = AC + 1 (AC = F(Y + 1) + 1)
3A6	6E15	SUB IP + 21	AC = AC – D вычитаем из результата ноль  AC = F(Y + 1) + 1 – 0
3A7	EE14	ST IP + 20	AC → D Сохраняем новый результат (D = F(Y + 1) + 1)
3A8	AE10	LD IP + 16	Z → AC
3A9	0740	DEC	AC = AC - 1 (AC = Z - 1)
3AA	0C00	PUSH	Передача Z – 1 (аргумента) в стек (0x7FF)
3AB	D71F	CALL 0x71F	Вызов подпрограммы F(Z – 1)

			(0x7FE)
3AC	0800	POP	Результат из SP → AC
3AD	0740	DEC	$AC = F(Z - 1) - 1$
3AE	6E0D	SUB IP + 13	$AC = F(Z - 1) - D$  $(AC = F(Z - 1) - F(Y + 1) - 1)$
3AF	EE0C	ST IP + 12	$AC \rightarrow D$  $(D = F(Z - 1) - F(Y + 1) - 1)$
3B0	AE0A	LD IP + 10	$X \rightarrow AC$
3B1	0700	INC	$AC = AC + 1$ $(AC = X + 1)$
3B2	0C00	PUSH	Передача аргумента $X + 1$ в стек (0x7FF)
3B3	D71F	CALL 0x71F	Вызов подпрограммы $F(X + 1)$ (0x7FE)
3B4	0800	POP	Результат из SP → AC
3B5	0740	DEC	$AC = AC - 1$  $(AC = F(X + 1) - 1)$
3B6	4E05	ADD IP + 5	$AC = AC + D$  $(AC = F(X + 1) - 1 + F(Z - 1) - F(Y + 1) - 1)$
3B7	EE04	ST IP + 4	$AC \rightarrow D$ , итоговый результат программы  $(D = F(X + 1) - 1 + F(Z - 1) - F(Y + 1) - 1)$
3B8	0100	HLT	Остановка программы

## Текст подпрограммы

Адрес	Код команды	Мнемоника	Комментарии
71F	AC01	LD &1	Загрузка аргумента (0x7FF) + флаги
720	F204	BMI IP + 4	Если <b>AC</b> < 0, переход на 0x725
721	F003	BEQ IP + 3	Если <b>AC</b> = 0, переход на 0x725
722	7E08	CMP IP + 8	Флаги по результату <b>AC</b> - <b>A</b>
723	F004	BEQ IP + 4	Если <b>AC</b> - <b>A</b> = 0, переход на 0x728
724	F803	BLT IP + 3	Если <b>AC</b> - <b>A</b> < 0, переход на 0x728
725	4C01	ADD &1	Умножить аргумент на 2
726	6E05	SUB IP + 5	<b>AC</b> = <b>AC</b> - <b>B</b> + флаги по результату.  т. е. Если изначальный аргумент был отрицательный, то <b>AC</b> = 2X - B
727	CE01	JUMP IP + 1	Переход на 0x729
728	AE02	LD IP + 2	<b>A</b> → <b>AC</b> -- Загрузка константы  т. е. Если значение <b>AC</b> <= <b>A</b> , загружаем <b>A</b> в <b>AC</b>
729	EC01	ST &1	<b>AC</b> → <b>M</b> Сохранение результата в ячейку аргумента программы в стеке
72A	0A00	RET	Возврат

## Данные подпрограммы

72B	0F6C	<b>A</b>	Константа
-----	------	----------	-----------

			<b>F6C (16) = 3948 (10)</b>
72C	00C1	B	<b>Константа, вычитаемое в 2X – B</b> <b>C1 (16) = 193 (10)</b>

## Данные

3B9	ZZZZ	Z	Значение Z
3BA	YYYY	Y	Значение Y
3BB	XXXX	X	Значение X
3BC	00C1	D	Результат

## Назначение программы, подпрограммы и комплекса программ и реализуемая ею функция

Программа вычисляет результат выражения:

$$D = F(x+1) - 1 + F(z-1) - F(y+1) - 1$$

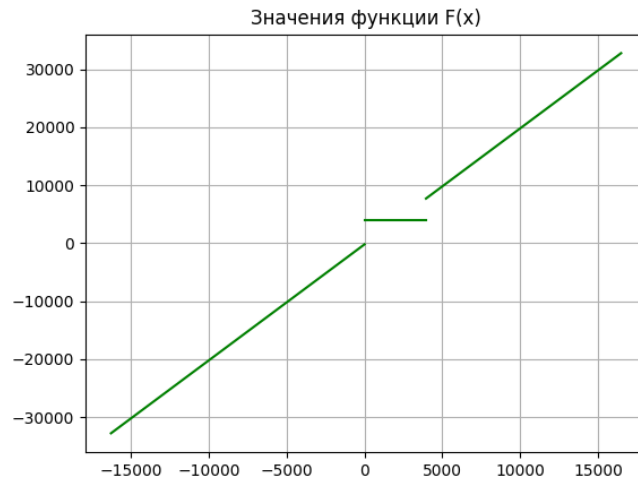
$$D = F(x+1) - F(y+1) + F(z-1) - 2$$

Программа трижды вызывает подпрограмму, передавая в неё различные значения аргументов:  $x+1$ ,  $y+1$  и  $z-1$  соответственно.

Подпрограмма вычисляет функцию, заданную следующим образом:

$$F(x) = \begin{cases} 2x - B; & x \notin (0, A] \\ A; & x \in (0, A] \end{cases}$$

Подпрограмма принимает аргумент через ячейку в стеке. Результат вычисления функции возвращается через ту же ячейку в стеке. То есть, одна и та же ячейка памяти используется и для передачи параметра в подпрограмму, и для возврата результата (в нашем случае, это 0x7FF).



## Область представления

Входные параметры  $X, Y, Z$  — 16-ти разрядные знаковые числа в допкоде.  $[-2^{15}, 2^{15}-1]$

Константы  $A, B$  — 16-ти разрядные знаковые числа в допкоде.  $[-2^{15}, 2^{15}-1]$

Результат  $D$  — 16-ти разрядное знаковое число в допкоде.  $[-2^{15}, 2^{15}-1]$

## Область допустимых значений

Область допустимых значений для функции, вычисляемой подпрограммой, относительно  $x$ .

Пусть

$$A = F6C_{16} = 3948_{10},$$

$$B = C1_{16} = 193_{10}$$

Тогда,

$$\begin{aligned} -2^{15} &\leq F(x) \leq 2^{15}-1, \\ -2^{15} &\leq 2x - 193 \leq 2^{15}-1, \\ -2^{15} + 193 &\leq 2x \leq 2^{15} + 192, \\ -16287 &\leq x \leq 16480. \end{aligned}$$

Область допустимых значений для итогового выражения, вычисляемого программой относительно  $X, Y, Z$ .

Учтём ограничения, налагаемые при вычислении итогового выражения программы.

$$\begin{aligned} -2^{15} &\leq D \leq 2^{15}-1 \\ -2^{15} &\leq F(X+1) - F(Y+1) + F(Z-1) - 2 \leq 2^{15}-1 \end{aligned}$$



$$-32766 \leq F(X+1) - F(Y+1) + F(Z-1) \leq 32769$$

Так как выражение является суммой результатов вызова подпрограммы, рассмотрим два худших случая с точки зрения переполнения при вычислении итогового выражения:

1. Переполнение положительных чисел (наиболее положительное значение выражения) —  $F(X+1)=2^{15}-1, F(Y+1)=-2^{15}, F(Z-1)=2^{15}-1$ ;
2. Переполнение отрицательных чисел (наиболее отрицательное значение выражения) —  $F(X+1)=-2^{15}, F(Y+1)=2^{15}-1, F(Z-1)=-2^{15}$ .

Тогда:

$$-32768 \leq \frac{F(X+1) - F(Y+1) + F(Z-1)}{3} \leq 32767,$$

$$-10923 \leq F(x) \leq 10922.$$

Тогда повторно вычислим допустимые значения аргумента функции с учётом новых ограничений:

$$-10923 \leq 2x - 193 \leq 10922,$$

$$-10730 \leq 2x \leq 11115,$$

$$-5365 \leq x \leq 5557.$$

*Область допустимых значений параметров программы (X, Y, Z).*

Так как параметры программы передаются в подпрограмму с инкрементом/декрементом их исходных значений, учтём это, дополнительно сузив область допустимых значений. То есть:

$$\begin{cases} -5365 \leq X+1 \leq 5557 \\ -5365 \leq Y+1 \leq 5557 \\ -5365 \leq Z-1 \leq 5557 \end{cases}$$

Итого, область допустимых значений параметров программы с учётом всех ограничений:

$$\begin{cases} -5366 \leq X \leq 5556 \\ -5366 \leq Y \leq 5556, \\ -5364 \leq Z \leq 5558 \end{cases}$$

$$-2^{15} \leq D \leq 2^{15} - 1.$$

## **Расположение данных в памяти ЭВМ программы, исходных данных и результатов:**

### **Программа**

0x39E — 0x3B8 - команды

0x3B9 — входное значение, переменная **Z** =90 (005A)

0x3BA — входное значение, переменная **Y** =5690 (163A)

0x3BB — входное значение, переменная **X** =-56 (FFC8)

0x3BC — Результат **D**

## Подпрограмма

0x71F — 0x72A - команды

0x72B — константа **A**

0x72C — константа **B**

## Адреса первой и последней выполняемой команды

### В программе

Адрес первой выполняемой команды: 0x39E

Адрес последней выполняемой команды: 0x3B8

### В подпрограмме

Адрес первой выполняемой команды: 0x71F

Адрес последней выполняемой команды: 0x72A

## Трассировка программы

Выполняемая команда		Содержимое регистров процессоров после выполнения команды								Ячейка, содержимое которой изменилось в ходе выполнения команды	
Адрес	Значение	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Значение
39E	0200	39F	0200	39E	0200	000	039E	0000	0100	-	-
39F	EE1C	3A0	EE1C	3BC	0000	000	001C	0000	0100	3BC	0000
3A0	AE19	3A1	AE19	3BA	163A	000	0019	163A	0000	-	-
3A1	0700	3A2	0700	3A1	0700	000	03A1	163B	0000	-	-
3A2	0C00	3A3	0C00	7FF	163B	7FF	03A2	163B	0000	7FF	163B
3A3	D71F	71F	D71F	7FE	03A4	7FE	D71F	163B	0000	7FE	03A4
71F	AC01	720	AC01	7FF	163B	7FE	0001	163B	0000	-	-
720	F204	721	F204	720	F204	7FE	0720	163B	0000	-	-
721	F003	722	F003	721	F003	7FE	0721	163B	0000	-	-
722	7E08	723	7E08	72B	0F6C	7FE	0008	163B	0001	-	-

723	F004	724	F004	723	F004	7FE	0723	163B	0001	-	-
724	F803	725	F803	724	F803	7FE	0724	163B	0001	-	-
725	4C01	726	4C01	7FF	163B	7FE	0001	2C76	0000	-	-
726	6E05	727	6E05	72C	00C1	7FE	0005	2BB5	0001	-	-
727	CE01	729	CE01	727	0729	7FE	0001	2BB5	0001	-	-
729	EC01	72A	EC01	7FF	2BB5	7FE	0001	2BB5	0001	7FF	2BB5
72A	0A00	3A4	0A00	7FE	03A4	7FF	072A	2BB5	0001	-	-
3A4	0800	3A5	0800	7FF	2BB5	000	03A4	2BB5	0001	-	-
3A5	0700	3A6	0700	3A5	0700	000	03A5	2BB6	0000	-	-
3A6	6E15	3A7	6E15	3BC	0000	000	0015	2BB6	0001	-	-
3A7	EE14	3A8	EE14	3BC	2BB6	000	0014	2BB6	0001	3BC	2BB6
3A8	AE10	3A9	AE10	3B9	005A	000	0010	005A	0001	-	-
3A9	0740	3AA	0740	3A9	0740	000	03A9	0059	0001	-	-
3AA	0C00	3AB	0C00	7FF	0059	7FF	03AA	0059	0001	7FF	0059
3AB	D71F	71F	D71F	7FE	03AC	7FE	D71F	0059	0001	7FE	03AC
71F	AC01	720	AC01	7FF	0059	7FE	0001	0059	0001	-	-
720	F204	721	F204	720	F204	7FE	0720	0059	0001	-	-
721	F003	722	F003	721	F003	7FE	0721	0059	0001	-	-
722	7E08	723	7E08	72B	0F6C	7FE	0008	0059	1000	-	-
723	F004	724	F004	723	F004	7FE	0723	0059	1000	-	-
724	F803	728	F803	724	F803	7FE	0003	0059	1000	-	-
728	AE02	729	AE02	72B	0F6C	7FE	0002	0F6C	0000	-	-
729	EC01	72A	EC01	7FF	0F6C	7FE	0001	0F6C	0000	7FF	0F6C
72A	0A00	3AC	0A00	7FE	03AC	7FF	072A	0F6C	0000	-	-
3AC	0800	3AD	0800	7FF	0F6C	000	03AC	0F6C	0000	-	-
3AD	0740	3AE	0740	3AD	0740	000	03AD	0F6B	0001	-	-
3AE	6E0D	3AF	6E0D	3BC	2BB6	000	000D	E3B5	1000	-	-
3AF	EE0C	3B0	EE0C	3BC	E3B5	000	000C	E3B5	1000	3BC	E3B5
3B0	AE0A	3B1	AE0A	3BB	FFC8	000	000A	FFC8	1000	-	-
3B1	0700	3B2	0700	3B1	0700	000	03B1	FFC9	1000	-	-
3B2	0C00	3B3	0C00	7FF	FFC9	7FF	03B2	FFC9	1000	7FF	FFC9
3B3	D71F	71F	D71F	7FE	03B4	7FE	D71F	FFC9	1000	7FE	03B4
71F	AC01	720	AC01	7FF	FFC9	7FE	0001	FFC9	1000	-	-
720	F204	725	F204	720	F204	7FE	0004	FFC9	1000	-	-
725	4C01	726	4C01	7FF	FFC9	7FE	0001	FF92	1001	-	-
726	6E05	727	6E05	72C	00C1	7FE	0005	FED1	1001	-	-

727	CE01	729	CE01	727	0729	7FE	0001	FED1	1001	-	-
729	EC01	72A	EC01	7FF	FED1	7FE	0001	FED1	1001	7FF	FED1
72A	0A00	3B4	0A00	7FE	03B4	7FF	072A	FED1	1001	-	-
3B4	0800	3B5	0800	7FF	FED1	000	03B4	FED1	1001	-	-
3B5	0740	3B6	0740	3B5	0740	000	03B5	FED0	1001	-	-
3B6	4E05	3B7	4E05	3BC	E3B5	000	0005	E285	1001	-	-
3B7	EE04	3B8	EE04	3BC	E285	000	0004	E285	1001	3BC	E285
3B8	0100	3B9	0100	3B8	0100	000	03B8	E285	1001	-	-

## Модернизированная программа

$D = F(x + 1) - F(y + 1) + F(z - 1) - 2;$

Необходимо модернизировать программу таким образом, чтобы она формировала 32-разрядный результат.

```

ORG 0x39E
START:
    CLA
    ST DH
    ST DL
fx:   LD X
    INC
    PUSH
    CALL $F
    POP
    BMI xmi
    ADD DL
    ST DL
    CLA
    ADC DH
    ST DH
    JUMP fy
xmi:  ADD DL
    ST DL
    LD #0xFF
    ADC DH
    ST DH
fy:   LD Y
    INC
    PUSH
    CALL $F
    POP
    NEG
    BMI ymi
    ADD DL
    ST DL
    CLA
    ADC DH
    ST DH
    JUMP fz
ymi:  ADD DL

```

```

        ST DL
        LD #0xFF
        ADC DH
        ST DH
fz:     LD Z
        DEC
        PUSH
        CALL $F
        POP
        BMI zmi
        ADD DL
        ST DL
        CLA
        ADC DH
        ST DH
        JUMP c
zmi:    ADD DL
        ST DL
        LD #0xFF
        ADC DH
        ST DH
c:      LD #2
        NEG
        ADD DL
        ST DL
        LD #0xFF
        ADC DH
        ST DH
        HLT
Z:      WORD 0x005A
Y:      WORD 0x163A
X:      WORD 0xFFC8
DL:     WORD 0
DH:     WORD 0

ORG 0x71F
F:      LD &1
        BMI LIN
        BEQ LIN
        CMP A
        BEQ CON
        BLT CON
LIN:    ADD &1
        SUB B
        JUMP EXIT
CON:    LD A
EXIT:   ST &1
        RET
A:      WORD 3948
B:      WORD 193

```

## **Вывод**

В ходе лабораторной работы были подробно изучены команды, в том числе связанные с работой со стеком: POP, PUSH, RET, CALL; принципы работы подпрограмм в БЭВМ, прямая адресация относительно SP.