

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

отчет
по лабораторной работе №3
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка целых чисел. Организация
ветвящихся процессов

Студент гр. 9382

Круглова В. Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться представлять и обрабатывать целые числа, а также организовывать ветвящиеся процесса

Основные теоретические положения.

Задание:

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет:

А) значения функций $i1 = f1(a, b, i)$ и $i2 = f2(a, b, i)$

Б) значения результирующей функции $res = f3(i1, i2, k)$, где вид функций $f1$ и $f2$ определяется из табл.2, а функция $f3$ – из табл.3 по цифрам шифра индивидуального задания ($n1, n2, n3$) приведенным в таблице 4.

Значения a, b, i, k являются исходными данными, которые должны выбираться студентами самостоятельно и задаваться в процессе исполнения программы в режиме отладки. При этом следует рассмотреть всевозможные комбинации параметров a, b и k , позволяющие проверить различные маршруты выполнения программы, а также различные знаки параметров a и b

Вариант 1

$f1 = \begin{cases} / 15-2*i, & \text{при } a > b \\ \end{cases}$

$\backslash 3*i+4, & \text{при } a \leq b$

$f2 = \begin{cases} / - (4*i+3), & \text{при } a > b \\ \end{cases}$

$\backslash 6*i - 10, & \text{при } a \leq b$

$f1 = \begin{cases} / \min(i1, i2), & \text{при } k = 0 \\ \end{cases}$

$\backslash \max(i1, i2), & \text{при } k \neq 0$

Ход работы:

В сегменте данных объявлены переменные a, b, i, k, i1, i2, res. Функции и ветвления реализованы через метки. Возвращаемые значения записываются в сегменте данных под соответствующей переменной или записываются в регистры. Для реализации ветвления использовалась команда CMP, она сравнивает два числа. В зависимости от результата сравнения, выполняется переход на ту или иную метку.

Исходный код программы.

```
STACKSG SEGMENT PARA STACK 'Stack'
        DW 32 DUP(?)
STACKSG ENDS

DATASG SEGMENT PARA 'Data' ;SEG DATA
    a DW -1h
    b DW 1h
    i DW 1h
    k DW -1h
    i1 DW 1h
    i2 DW 1h
    res DW 1h
DATASG ENDS ;ENDS DATA

CODE SEGMENT ;SEG CODE
ASSUME DS:DATASG, CS:CODE

Main PROC FAR
    mov ax, DATASG
    mov ds, ax

f1:
    mov ax, a ; переменная a в ax
    cmp ax, b ; сравниваем переменные a и b соответственно
    jle f1_jle ; a <= b

    ; если попали сюда, то a > b
    mov ax, i ; переменная i в ax
    shl ax, 1 ; умножим i на 2
    mov bx, 15 ; кладем в bx 15
    sub bx, ax ; bx - ax = 15 - i * 2
    mov i1, bx ; записываем в i1 результат 15 - i * 2
    jmp f2 ; переходим к f2

f1_jle: ; a <= b
    mov ax, i ; переменная i в ax
    mov bx, i ; переменная i в bx
    shl ax, 1 ; домножим ax на 2
    shl ax, 1 ; домножим ax на 2
    shl bx, 1 ; домножим bx на 2
    add ax, bx ; прибавим к ax bx, получим 6 * i
    shr ax, 1 ; поделим ax на 2
```

```

    mov bx, 4h          ; кладем в bx 4
    add ax, bx          ; ax + bx = 3 * i + 4
    mov i1, ax          ; записываем в i1 результат 3 * i + 4
    jmp f2

f2:
    mov ax, a           ; переменная a в ax
    cmp ax, b           ; сравниваем переменные a и b
    jle f2_jle          ; a <= b

                        ; если попали сюда, то a > b (ja)

    mov ax, i           ; переменная i в ax
    mov bx, i           ; переменная i в bx
    shl ax, 1           ; домножим ax на 2
    shl ax, 1           ; домножим ax на 2
    mov bx, 3h          ; кладем в bx 3
    add ax, bx          ; ax = 4 * i + 3
    mov bx, 0h          ; кладем в bx 0
    sub bx, ax          ; bx = -(4 * i + 3)
    mov i2, bx          ; записываем в i2 результат (1 + i) * 2 - 4
    jmp f3              ; переходим к f3

f2_jle:
    mov ax, i           ; переменная i в ax
    mov bx, i           ; переменная i в bx
    shl ax, 1           ; домножим ax на 2
    shl ax, 1           ; домножим ax на 2
    shl bx, 1           ; домножим bx на 2
    add ax, bx          ; прибавим к ax bx, получим 6 * i
    mov bx, 10          ; кладем в bx 10
    sub ax, bx          ; ax - bx = 6 * i - 10
    mov i2, ax          ; записываем в i2 результат 6 * i - 10
    jmp f3

f3:
    mov ax, k           ; кладем в ax переменную k
    cmp ax, 0           ; сравним k с 0
    je f3_je            ; k = 0

                        ; если оказались здесь, то k < 0

    mov ax, i1          ; кладем в ax переменную i1
    mov bx, i2          ; кладем в ax переменную i1

    cmp ax, bx          ; сравним i1 с i2

    jle f3_jle          ; если i1 <= i2
    jmp f3_jl_result_jge

f3_jle:
    mov ax, i2          ; кладем в ax переменную i2
    jmp f3_jl_result_jge

f3_je:
    mov ax, i1          ; кладем в ax переменную i1
    mov bx, i2          ; кладем в ax переменную i1

```

```

    cmp ax, bx          ; сравним i1 с i2
    jge f3_jge          ; если i1 >= i2
    jmp f3_jl_result_jge

f3_jge:
    mov ax, i2          ; кладем в ax переменную i2
    jmp f3_jl_result_jge

f3_jl_result_jge:
    mov res, ax         ; максимум или минимум в зависимости от условия
    jmp end_f           ; завершаем программу

end_f:
    mov ah, 4ch         ; и наконец завершим программу
    int 21h

Main      ENDP
CODE      ENDS
END Main                                     ;ENDS CODE

```

Листинг программы.

#Microsoft (R) Macro Assembler Version 5.10 10/22/20 11:10:1

Page 1-1

```

0000          STACKSG SEGMENT PARA STACK 'Stack'
0000 0020[          DW    32 DUP(?)
          ????)
          ]

```

```

0040          STACKSG  ENDS

```

```

0000          DATASG SEGMENT PARA 'Data'
          ;SEG DATA

```

```

0000 FFFF          a      DW -1h
0002 0001          b      DW 1h
0004 0001          i      DW 1h
0006 FFFF          k      DW -1h
0008 0001          i1     DW 1h
000A 0001          i2     DW 1h

```

```

000C 0001          res    DW 1h
000E              DATASG    ENDS

                                ;ENDS DATA

0000              CODE     SEGMENT

                                ;SEG CODE
                                ASSUME DS:DATASG, CS:CODE

0000              Main    PROC FAR
0000 B8 ---- R          mov ax, DATASG
0003 8E D8            mov ds, ax

0005              f1:
0005 A1 0000 R          mov ax, a      ; Ð¿Ð¼Ñ Ð¼ÐÐÐÐ¼ÐœÐœÐ°Ñ a
                                Ð² ax
0008 3B 06 0002 R      cmp ax, b      ; Ñ Ñ Ð°Ð²ÐœÐŹÐ²Ð°Ð¼ÐÐÐ Ð
                                ¿Ð¼Ñ Ð¼ÐÐÐÐ¼ÐœÐœÑ Ð¼ a ÐŹ b
Ñ ÐŸÐŸÑ Ð²Ð¼Ñ Ñ Ñ Ð²
                                Ð¼ÐœÐœÐŸ
000C 7E 11            jle f1_jle      ; a <= b

                                ; Ð¼Ñ Ð»ÐŹ Ð¿ÐŸÐ¿Ð°Ð»ÐŹ
Ñ Ñ ÐŹÐ°, Ñ ÐŸ a > b
000E A1 0004 R          mov ax, i      ; Ð¿Ð¼Ñ Ð¼ÐÐÐÐ¼ÐœÐœÐ°Ñ i
                                Ð² ax
0011 D1 E0            shl ax, 1      ; Ñ ÐœÐœÐŸÐŸÐŹÐœ i ÐœÐ°
                                2
0013 BB 000F          mov bx, 15      ; Ð°Ð»Ð°ÐŹÐ¼ÐÐÐ Ð² bx 15
0016 2B D8            sub bx, ax      ; bx - ax = 15 - i * 2
0018 89 1E 0008 R      mov i1, bx     ; Ð°Ð¿Ð¿ÐŹÑ Ñ Ð²Ð°Ð¼ÐÐÐ
² i1 Ñ Ð¼Ð·Ñ Ð»Ñ Ñ Ð°Ñ 15 - i * 2
001C EB 1D 90          jmp f2         ; Ð¿Ð¼Ñ Ð¼Ñ ÐŸÐŹÐŹÐœ Ð°
f2

001F              f1_jle:            ; a <= b

```

001F A1 0004 R mov ax, i ; $\text{D}_i \text{D}_{\mu} \tilde{\text{N}} \text{D}_{\mu} \text{DCE} \text{D}_{\mu} \text{DCE} \text{DCE} \text{D}^{\circ} \tilde{\text{N}} \text{ i}$
 $\text{D}^2 \text{ ax}$

0022 8B 1E 0004 R mov bx, i ; $\text{D}_i \text{D}_{\mu} \tilde{\text{N}} \text{D}_{\mu} \text{DCE} \text{D}_{\mu} \text{DCE} \text{DCE} \text{D}^{\circ} \tilde{\text{N}} \text{ i}$
 $\text{D}^2 \text{ bx}$

0026 D1 E0 shl ax, 1 ; $\text{D} \tilde{\text{Z}} \text{D} \tilde{\text{Y}} \text{DCE} \text{DCE} \text{D} \tilde{\text{Y}} \text{D} \text{D} \tilde{\text{Z}} \text{DCE} \text{ ax DCE}$

#Microsoft (R) Macro Assembler Version 5.10 10/22/20 11:10:1

Page 1-2

$\text{D}^{\circ} 2$

0028 D1 E0 shl ax, 1 ; $\text{D} \tilde{\text{Z}} \text{D} \tilde{\text{Y}} \text{DCE} \text{DCE} \text{D} \tilde{\text{Y}} \text{D} \text{D} \tilde{\text{Z}} \text{DCE} \text{ ax DCE}$
 $\text{D}^{\circ} 2$

002A D1 E3 shl bx, 1 ; $\text{D} \tilde{\text{Z}} \text{D} \tilde{\text{Y}} \text{DCE} \text{DCE} \text{D} \tilde{\text{Y}} \text{D} \text{D} \tilde{\text{Z}} \text{DCE} \text{ bx DCE}$
 $\text{D}^{\circ} 2$

002C 03 C3 add ax, bx ; $\text{D}_i \tilde{\text{N}} \text{D} \tilde{\text{Z}} \text{D} \pm \text{D}^{\circ} \text{D}^2 \text{D} \tilde{\text{Z}} \text{DCE} \text{D}^{\circ} \text{ ax}$
 $\text{bx, D}_i \text{D} \tilde{\text{Y}} \text{D} \gg \tilde{\text{N}} \tilde{\text{N}} \text{D} \tilde{\text{Z}} \text{DCE} 6 * i$

002E D1 E8 shr ax, 1 ; $\text{D}_i \text{D} \tilde{\text{Y}} \text{D} \tilde{\text{Z}} \text{D}_{\mu} \text{D} \gg \text{D} \tilde{\text{Z}} \text{DCE} \text{ ax DCE} \text{D}^{\circ}$
 2

0030 BB 0004 mov bx, 4h ; $\text{D}^{\circ} \text{D} \gg \text{D}^{\circ} \text{D} \tilde{\text{Z}} \text{D}_{\mu} \text{DCE} \text{D}^2 \text{ bx } 4$

0033 03 C3 add ax, bx ; $\text{ax} + \text{bx} = 3 * i + 4$

0035 A3 0008 R mov i1, ax ; $\text{D} \cdot \text{D}^{\circ} \text{D}_i \text{D} \tilde{\text{Z}} \tilde{\text{N}} \tilde{\text{N}} \text{D}^2 \text{D}^{\circ} \text{D}_{\mu} \text{DCE} \text{D}$
 $^2 i1 \tilde{\text{N}} \text{D}_{\mu} \text{D} \cdot \tilde{\text{N}} \text{D} \gg \tilde{\text{N}} \tilde{\text{N}} \text{D}^{\circ} \tilde{\text{N}} 3 * i + 4$

0038 EB 01 90 jmp f2

003B f2:

003B A1 0000 R mov ax, a ; $\text{D}_i \text{D}_{\mu} \tilde{\text{N}} \text{D}_{\mu} \text{DCE} \text{D}_{\mu} \text{DCE} \text{DCE} \text{D}^{\circ} \tilde{\text{N}} \text{ a}$
 $\text{D}^2 \text{ ax}$

003E 3B 06 0002 R cmp ax, b ; $\tilde{\text{N}} \tilde{\text{N}} \text{D}^{\circ} \text{D}^2 \text{DCE} \text{D} \tilde{\text{Z}} \text{D}^2 \text{D}^{\circ} \text{D}_{\mu} \text{DCE} \text{D}$
 $\text{D}_i \text{D}_{\mu} \tilde{\text{N}} \text{D}_{\mu} \text{DCE} \text{D}_{\mu} \text{DCE} \text{DCE} \tilde{\text{N}} \text{D}_{\mu} \text{a D} \tilde{\text{Z}} \text{b}$

0042 7E 1C jle f2_jle ; $\text{a} \leq \text{b}$

; $\text{D}_{\mu} \tilde{\text{N}} \text{D} \gg \text{D} \tilde{\text{Z}} \text{D}_i \text{D} \tilde{\text{Y}} \text{D}_i \text{D}^{\circ} \text{D} \gg \text{D} \tilde{\text{Z}}$

$\tilde{\text{N}} \tilde{\text{N}} \text{D} \tilde{\text{Z}} \text{D}^{\circ}, \tilde{\text{N}} \text{D} \tilde{\text{Y}} \text{a} > \text{b} (\text{ja})$

0044 A1 0004 R mov ax, i ; $\text{D}_i \text{D}_{\mu} \tilde{\text{N}} \text{D}_{\mu} \text{DCE} \text{D}_{\mu} \text{DCE} \text{DCE} \text{D}^{\circ} \tilde{\text{N}} \text{ i}$

		$\text{D}^2 \text{ ax}$
0047 8B 1E 0004 R	mov bx, i	; $\text{D}_i \text{D}_{\mu\tilde{\text{N}}} \text{D}_{\mu\text{DCE}} \text{D}_{\mu\text{D}\alpha\text{D}\alpha\text{D}^\circ\tilde{\text{N}}} \text{ i}$
		$\text{D}^2 \text{ bx}$
004B D1 E0	shl ax, 1	; $\text{D}\check{\text{Z}}\text{D}\check{\text{Y}}\text{DCE}\text{D}\alpha\text{D}\check{\text{Y}}\text{D}\P\text{D}\check{\text{Z}}\text{DCE} \text{ ax D}\alpha$
		$\text{D}^\circ 2$
004D D1 E0	shl ax, 1	; $\text{D}\check{\text{Z}}\text{D}\check{\text{Y}}\text{DCE}\text{D}\alpha\text{D}\check{\text{Y}}\text{D}\P\text{D}\check{\text{Z}}\text{DCE} \text{ ax D}\alpha$
		$\text{D}^\circ 2$
004F BB 0003	mov bx, 3h	; $\text{D}^\circ\text{D}\gg\text{D}^\circ\text{D}\check{\text{Z}}\text{D}_{\mu\text{DCE}} \text{ D}^2 \text{ bx } 3$
0052 03 C3	add ax, bx	; $\text{ax} = 4 * \text{i} + 3$
0054 BB 0000	mov bx, 0h	; $\text{D}^\circ\text{D}\gg\text{D}^\circ\text{D}\check{\text{Z}}\text{D}_{\mu\text{DCE}} \text{ D}^2 \text{ bx } 0$
0057 2B D8	sub bx, ax	; $\text{bx} = -(4 * \text{i} + 3)$
0059 89 1E 000A R	mov i2, bx	; $\text{D}\cdot\text{D}^\circ\text{D}_i\text{D}\check{\text{Z}}\tilde{\text{N}} \tilde{\text{N}} \text{ D}^2\text{D}^\circ\text{D}_{\mu\text{DCE}} \text{ D}$
		$^2 \text{i}2 \tilde{\text{N}} \text{ D}_{\mu\text{D}}\cdot\tilde{\text{N}} \text{ D}\gg\tilde{\text{N}} \tilde{\text{N}} \text{ D}^\circ\tilde{\text{N}} (1 + \text{i}) * 2 - 4$
005D EB 1B 90	jmp f3	; $\text{D}_i\text{D}_{\mu\tilde{\text{N}}} \text{D}_{\mu\tilde{\text{N}}} \text{D}\check{\text{Y}}\text{D}\check{\text{Z}}\text{D}\check{\text{Z}}\text{DCE} \text{ D}^\circ$
		f3
0060	f2_jle:	
0060 A1 0004 R	mov ax, i	; $\text{D}_i\text{D}_{\mu\tilde{\text{N}}} \text{D}_{\mu\text{DCE}} \text{D}_{\mu\text{D}\alpha\text{D}\alpha\text{D}^\circ\tilde{\text{N}}} \text{ i}$
		$\text{D}^2 \text{ ax}$
0063 8B 1E 0004 R	mov bx, i	; $\text{D}_i\text{D}_{\mu\tilde{\text{N}}} \text{D}_{\mu\text{DCE}} \text{D}_{\mu\text{D}\alpha\text{D}\alpha\text{D}^\circ\tilde{\text{N}}} \text{ i}$
		$\text{D}^2 \text{ bx}$
0067 D1 E0	shl ax, 1	; $\text{D}\check{\text{Z}}\text{D}\check{\text{Y}}\text{DCE}\text{D}\alpha\text{D}\check{\text{Y}}\text{D}\P\text{D}\check{\text{Z}}\text{DCE} \text{ ax D}\alpha$
		$\text{D}^\circ 2$
0069 D1 E0	shl ax, 1	; $\text{D}\check{\text{Z}}\text{D}\check{\text{Y}}\text{DCE}\text{D}\alpha\text{D}\check{\text{Y}}\text{D}\P\text{D}\check{\text{Z}}\text{DCE} \text{ ax D}\alpha$
		$\text{D}^\circ 2$
006B D1 E3	shl bx, 1	; $\text{D}\check{\text{Z}}\text{D}\check{\text{Y}}\text{DCE}\text{D}\alpha\text{D}\check{\text{Y}}\text{D}\P\text{D}\check{\text{Z}}\text{DCE} \text{ bx D}\alpha$
		$\text{D}^\circ 2$
006D 03 C3	add ax, bx	; $\text{D}_i\tilde{\text{N}} \text{ D}\check{\text{Z}}\text{D}\pm\text{D}^\circ\text{D}^2\text{D}\check{\text{Z}}\text{DCE} \text{ D}^\circ \text{ ax}$


```

bx, Ð¿ÐŸÐ»Ñ Ñ ÐžÐœ 6 * i
006F BB 000A          mov bx, 10      ; Ð°Ð»ÐžÐµÐ² bx 10
0072 2B C3          sub ax, bx      ; ax - bx = 6 * i - 10
0074 A3 000A R      mov i2, ax      ; Ð·Ð°Ð¿ÐžÑ Ñ Ð²Ð°ÐµÐµ Ð
² i2 Ñ ÐµÐ·Ñ Ð»Ñ Ñ Ð°Ñ 6 * i - 10
0077 EB 01 90          jmp f3

007A                f3:
007A A1 0006 R      mov ax, k      ; Ð°Ð»ÐžÐµÐ² ax Ð¿Ð
µÑ ÐµÐœÐµÐœÑ Ñ k
007D 3D 0000          cmp ax, 0      ; Ñ Ñ Ð°Ð²ÐœÐžÐœ k Ñ 0
0080 74 14          je f3_je      ; k = 0

                ; ÐµÑ Ð»Ðž ÐŸÐ°Ð·Ð°Ð»Ð»
žÑ Ñ Ð·ÐžÐµÑ Ñ , Ñ Ÿ k < 0
0082 A1 0008 R      mov ax, i1      ; Ð°Ð»ÐžÐµÐ² ax Ð¿Ð
µÑ ÐµÐœÐµÐœÑ Ñ i1
0085 8B 1E 000A R    mov bx, i2      ; Ð°Ð»ÐžÐµÐ² bx Ð¿Ð
µÑ ÐµÐœÐµÐœÑ Ñ i1

0089 3B C3          cmp ax, bx      ; Ñ Ñ Ð°Ð²ÐœÐžÐœ i1 c i2

008B 7E 03          jle f3_jle      ; ÐµÑ Ð»Ðž i1 <= i2
008D EB 1B 90          jmp f3_jl_result_jge

0090                f3_jle:
0090 A1 000A R      mov ax, i2      ; Ð°Ð»ÐžÐµÐ² ax Ð¿Ð
µÑ ÐµÐœÐµÐœÑ Ñ i2
0093 EB 15 90          jmp f3_jl_result_jge

```

0096	f3_je:
0096 A1 0008 R	mov ax, i1 ; Ð°Ð»Ð°ŽÐµÐ£ Ð² ax Ð¿ÐµÑ ÐµÐ£ÐµÐ£Ð£Ñ Ñ i1
0099 8B 1E 000A R	mov bx, i2 ; Ð°Ð»Ð°ŽÐµÐ£ Ð² ax Ð¿ÐµÑ ÐµÐ£ÐµÐ£Ð£Ñ Ñ i1
009D 3B C3	cmp ax, bx ; Ñ Ñ Ð°Ð²Ð£Ð¿Ð£ i1 c i2
009F 7D 03	jge f3_jge ; ÐµÑ Ð»Ð¿ i1 >= i2
00A1 EB 07 90	jmp f3_jl_result_jge
00A4	f3_jge:
00A4 A1 000A R	mov ax, i2 ; Ð°Ð»Ð°ŽÐµÐ£ Ð² ax Ð¿ÐµÑ ÐµÐ£ÐµÐ£Ð£Ñ Ñ i2
00A7 EB 01 90	jmp f3_jl_result_jge
00AA	f3_jl_result_jge:
00AA A3 000C R	mov res, ax ; Ð£Ð°Ð°Ñ Ð¿Ð£Ñ Ð£ Ð¿Ð»Ð
	Ž Ð£Ð¿Ð£Ð¿Ð£Ñ Ð£ Ð²
	Ð·Ð²Ð¿Ñ Ð¿Ð£ÐŸÑ Ñ Ð¿ ÐŸÑ
	Ñ Ñ Ð»ÐŸÐ²Ð¿Ñ
00AD EB 01 90	jmp end_f ; Ð·Ð²Ð²ÐµÑ Ñ Ð°ÐµÐ£ Ð¿Ñ

ĐŸĐ³Ñ Đ°ĐÆĐÆÑ

```
00B0          end_f:
00B0 B4 4C      mov ah, 4ch      ; Đž ĐæĐ°Đ°ĐŸĐæĐµÑ Đ·Đ°
                Đ²ĐµÑ Ñ ĐžĐÆ Đ¿Ñ ĐŸĐ³Ñ Đ°ĐÆĐÆÑ
00B2 CD 21      int 21h

00B4          Main   ENDP
00B4          CODE   ENDS
                END Main                ;ENDS C
                ODE
```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
CODE	00B4	PARA	NONE	
DATASG	000E	PARA	NONE	'DATA'
STACKSG	0040	PARA	STACK	'STACK'

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATASG
B	L WORD	0002	DATASG
END_F	L NEAR	00B0	CODE
F1	L NEAR	0005	CODE
F1_JLE	L NEAR	001F	CODE
F2	L NEAR	003B	CODE
F2_JLE	L NEAR	0060	CODE
F3	L NEAR	007A	CODE
F3_JE	L NEAR	0096	CODE
F3_JGE	L NEAR	00A4	CODE
F3_JLE	L NEAR	0090	CODE
F3_JL_RESULT_JGE	L NEAR	00AA	CODE
I	L WORD	0004	DATASG
I1	L WORD	0008	DATASG
I2	L WORD	000A	DATASG

```

K ..... L WORD 0006 DATASG

MAIN ..... F PROC 0000 CODE Length = 00B4

RES ..... L WORD 000C DATASG

@CPU ..... TEXT 0101h
@FILENAME ..... TEXT ZXC
@VERSION ..... TEXT 510

```

122 Source Lines

122 Total Lines

28 Symbols

48000 + 426521 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Тестирование.

№	Входные данные	Выходные данные	Правильный результат
1	a=1, b=1, i=1, k=1	i1=7, i2=-4, res=7	i1=7, i2=-4, res=7
2	a=1, b=1, i=1, k=-1	i1=7, i2=-4, res=7	i1=7, i2=-4, res=7
3	a=2, b=1, i=1, k=1	i1=13, i2=-7, res=13	i1=13, i2=-7, res=13
4	a=2, b=1, i=1, k=0	i1=13, i2=-7, res=-7	i1=13, i2=-7, res=-7
5	a=-1, b=1, i=1, k=0	i1=7, i2=-4, res=-4	i1=7, i2=-4, res=-4
6	a=-1, b=1, i=1, k=0	i1=7, i2=-4, res=-4	i1=7, i2=-4, res=-4

Выводы.

Были изучены режимы адресации, определены ошибки в программе, и было дано объяснение ошибкам.