

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

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Организация ЭВМ и систем»**  
**Представление и обработка целых чисел. Организация ветвящихся**  
**процессов**

Студент гр. 0383

\_\_\_\_\_

Девятериков И.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

## Вариант № 2.

### Цель работы.

Изучение режима адресации при помощи программы, которая тестирует режимы адресации. Научиться находить допущенные при реализации ошибки адресации.

### Задание.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров  $a$ ,  $b$ ,  $i$ ,  $k$  вычисляет: значения функций  $i1 = f1(a,b,i)$  и  $i2 = f2(a,b,i)$ ; значения результирующей функции  $res = f3(i1,i2,k)$ , где вид функций  $f1$  и  $f2$  определяется из табл. 2, а функции  $f3$  - из табл.3 по цифрам шифра индивидуального задания ( $n1,n2,n3$ ), приведенным в табл.4. Значения  $a$ ,  $b$ ,  $i$ ,  $k$  являются исходными данными, которые должны выбираться студентом 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  $b$ .

$$f1 = \begin{cases} / 15-2*i, & \text{при } a > b \\ \backslash 3*i+4, & \text{при } a \leq b \end{cases}$$

$$f3 = \begin{cases} / 7 - 4*i, & \text{при } a > b \\ \backslash 8 - 6*i, & \text{при } a \leq b \end{cases}$$

$$f2 = \begin{cases} / \max(i1, 10-i2), & \text{при } k < 0 \\ \backslash |i1 - i2|, & \text{при } k \geq 0 \end{cases}$$

### Выполнение работы.

В ходе выполнения работы были использованы команды `str`, `jle/jg`, `jde/jl`, выполняющие сравнение значений, позволяющие выполнять только определённые участки кода.

Все возможные варианты работы кода в зависимости от исходных данных представлены в Табл.1.

Табл.1

Значения a, b, i, k	Результат вычисления f1	Результат вычисления f3	Результат вычисления f2	Комментарий
a = -1 b = 1 i = 4 k = -1	0010 = 16	FFF0 = -16	001A = 26	Верно
a = -1 b = 1 i = 4 k = 1	0010 = 16	FFF0 = -16	0020 = 32	Верно
a = 1 b = -1 i = 4 k = -1	0007 = 7	FFF7 = -9	0013 = 19	Верно
a = 1 b = -1 i = 4 k = 1	0007 = 7	FFF7 = -9	0010 = 16	Верно

Разработанный программный код см. **Приложение А.**

**Выводы.**

В ходе выполнения лабораторной работы были изучены различные способы сравнения чисел. Был изучен процесс работы программы с ветвлением в процессе выполнения.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

main.asm

AStack SEGMENT STACK

    DW 12 DUP(?)

AStack ENDS

;Данные программы

DATA     SEGMENT

;Директивы описания данных

a     DW   0

b     DW   0

i     DW   0

k     DW   0

i1    DW   0

i2    DW   0

DATA     ENDS

; Код программы

CODE     SEGMENT

    ASSUME CS:CODE, DS:DATA, SS:AStack

; Главная процедура

Main     PROC FAR

    push DS

    sub  AX,AX

    push AX

    mov  AX,DATA

    mov  DS,AX

```
mov a, 1
mov b, -1
mov i, 4
mov k, 1
```

```
;f1&f3
mov cx, i
shl cx, 1 ; cx = 2*i
mov ax, cx
mov bx, b
```

```
cmp a, bx ; сравнение a и b
jle f1f3
```

```
    neg cx ; a > b
    add cx, 15
    mov i1, cx
```

```
    shl ax, 1
    neg ax
    add ax, 7
    mov i2, ax
    jmp f1f3End
```

```
f1f3: ; a <= b
    add cx, i
    add cx, 4
    mov i1, cx
```

```

        mov cx, ax
        shl ax, 1
        add ax, cx
        neg ax
        add ax, 8
        mov i2, ax
f1f3End:

;f2
mov ax, i1
mov cx, i2
neg cx
mov bx, k

cmp bx, 0
jge f2
    add cx, 10 ; k < 0
    cmp ax, cx
    jge max
        mov ax, cx
    max: ; ax >= cx
    jmp f2End
f2: ; k >= 0
    add ax, cx
    cmp ax, 0
    jge modul
        neg ax
    modul: ; ax >= 0
f2End:

```

MainEnd: ; в ax лежит значение функции f2

ret

Main ENDP

CODE ENDS

END Main

main.lst



```
0000          AStack SEGMENT STACK
0000 000C[          DW 12 DUP(?)
          ????
          ]

0018          AStack ENDS
          ;Данные программы
0000          DATA  SEGMENT
          ;Директивы описания данных
0000 0000          a   DW  0
0002 0000          b   DW  0
0004 0000          i   DW  0
0006 0000          k   DW  0
0008 0000          i1  DW  0
000A 0000          i2  DW  0

000C          DATA  ENDS

          ; Код программы
0000          CODE   SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

          ; Головная процедура
0000          Main   PROC FAR
0000 1E          push DS
```

0001 2B C0	sub AX,AX
0003 50	push AX
0004 B8 ---- R	mov AX,DATA
0007 8E D8	mov DS,AX
0009 C7 06 0000 R FFFF	mov a, -1
000F C7 06 0002 R 0001	mov b, 1
0015 C7 06 0004 R 0004	mov i, 4
001B C7 06 0006 R FFFF	mov k, -1
	;f1&f3
0021 8B 0E 0004 R	mov cx, i
0025 D1 E1	shl cx, 1 ; cx = 2*i
0027 8B C1	mov ax, cx
0029 8B 1E 0002 R	mov bx, b
002D 39 1E 0000 R	cmp a, bx ; сравнение a и b
0031 7E 16	jle f1f3
0033 F7 D9	neg cx ; a > b
0035 83 C1 0F	add cx, 15
0038 89 0E 0008 R	mov i1, cx
003C D1 E0	shl ax, 1
003E F7 D8	neg ax
0040 05 0007	add ax, 7
0043 A3 000A R	mov i2, ax
0046 EB 1A 90	jmp f1f3End
0049	f1f3: ; a <= b



```
0049 03 0E 0004 R      add cx, i
004D 83 C1 04          add cx, 4
0050 89 0E 0008 R      mov i1, cx

0054 8B C8            mov cx, ax
0056 D1 E0            shl ax, 1
0058 03 C1            add ax, cx
005A F7 D8            neg ax
005C 05 0008          add ax, 8
005F A3 000A R        mov i2, ax
0062                  f1f3End:

;f2
0062 A1 0008 R        mov ax, i1
0065 8B 0E 000A R      mov cx, i2
0069 F7 D9            neg cx
006B 8B 1E 0006 R      mov bx, k

006F 83 FB 00          cmp bx, 0
0072 7D 0C            jge f2
0074 83 C1 0A          add cx, 10 ; k < 0
0077 3B C1            cmp ax, cx
0079 7D 02            jge max
007B 8B C1            mov ax, cx
007D                  max: ; ax >= cx
```

007D	EB 0A 90	jmp f2End
0080		f2: ; k >= 0
0080	03 C1	add ax, cx
0082	3D 0000	cmp ax, 0
0085	7D 02	jge modul
0087	F7 D8	neg ax
0089		modul: ; ax >= 0
0089		f2End:
0089		MainEnd: ; в ax лежит значение функции f2
0089	CB	ret
008A		Main ENDP
008A		CODE ENDS
		END Main

## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK .....	0018	PARA		STACK
CODE .....	008A	PARA		NONE
DATA .....	000C	PARA		NONE

## Symbols:

N a m e	Type	Value	Attr
A .....	L WORD	0000	DATA
B .....	L WORD	0002	DATA
F1F3 .....	L NEAR	0049	CODE
F1F3END .....	L NEAR	0062	CODE
F2 .....	L NEAR	0080	CODE
F2END .....	L NEAR	0089	CODE
I .....	L WORD	0004	DATA
I1 .....	L WORD	0008	DATA
I2 .....	L WORD	000A	DATA

K ..... L WORD 0006 DATA

MAIN ..... F PROC 0000 CODE Length =  
008A

MAINEND ..... L NEAR 0089 CODE

MAX ..... L NEAR 007D CODE

MODUL ..... L NEAR 0089 CODE

@CPU ..... TEXT 0101h

@FILENAME ..... TEXT main

@VERSION ..... TEXT 510

89 Source Lines

89 Total Lines

22 Symbols

47966 + 461341 Bytes symbol space free

0 Warning Errors

0 Severe Errors