

**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»**  
**Тема: Представление и обработка целых чисел. Организация ветвящихся**  
**процессов**

Студентка гр. 1381

\_\_\_\_\_

Тулегенова А.О.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить представление и обработку целых чисел, научиться организовывать ветвящиеся процессы.

### **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 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  $b$ .

Вариант 25:

$$\begin{aligned} f5 &= \begin{cases} 20 - 4i, & \text{при } a > b \\ -(6i - 6), & \text{при } a \leq b \end{cases} \\ f8 &= \begin{cases} -(6i+8), & \text{при } a > b \\ 9 - 3(i-1), & \text{при } a \leq b \end{cases} \\ f6 &= \begin{cases} |i1 - i2|, & \text{при } k < 0 \\ \max(7, |i2|), & \text{при } k \geq 0 \end{cases} \end{aligned}$$

### **Ход выполнения работы**

Были заданы сегменты данных под ДОС, определена модель памяти small, отведена память под размер стека, проинициализированы переменные  $a, b, i, k$ . Далее было загружено в  $ds$  начало сегмента данных, происходит сравнение переменных  $a$  и  $b$  с помощью команды `cmp`. Если  $a > b$ , то с помощью команды `jg` выполняется переход к метке `func2`, иначе выполняется метка `func1`.

Далее происходит вычисление значений  $i1$  и  $i2$  с помощью команд арифметического сдвига, сложения и вычитания, данные значения записываются в регистры  $ax$  и  $sx$ . Далее переход к метке `func3`, значение переменной  $k$  сравнивается с нулем и если он больше или равен нулю, то происходит переход к метке `max`, в которой берется по модулю значение  $i2$  и оно сравнивается с 7, большее значение записывается в регистр  $sx$ . Если значение переменной  $k$  меньше нуля, то происходит переход к метке `abs`, в которой в регистр  $sx$  записывается модуль разности между значениями  $i1$  и  $i2$ . Далее переходим к метке `result`, в переменную `res` записывается значение регистра  $sx$ . После этого программа завершается.

Таблица 1. Тестирование работы программы.

Значение констант	Ожидаемый результат	Полученный результат
$a = 1$ $b = 2$ $i = 3$ $k = 4$	$i1 = -12$ (FFF4) $i2 = 3$ (0003) $f3 = 7$ (0007)	$i1 = -12$ (FFF4) $i2 = 3$ (0003) $f3 = 7$ (0007)
$a = 3$ $b = 1$ $i = -1$ $k = 0$	$i1 = 24$ (0018) $i2 = -2$ (FFFE) $f3 = 7$ (0007)	$i1 = 24$ (0018) $i2 = -2$ (FFFE) $f3 = 7$ (0007)
$a = 2$ $b = 2$ $i = 0$ $k = -1$	$i1 = 6$ (0006) $i2 = 12$ (000C) $f3 = 6$ (0006)	$i1 = 6$ (0006) $i2 = 12$ (000C) $f3 = 6$ (0006)

## Вывод

При выполнении лабораторной работы было изучено представление и обработка целых чисел, организация ветвящихся процессов.

## ПРИЛОЖЕНИЕ А

### КОД ПРОГРАММЫ

Название файла: lab3.asm

```
dosseg
.model small
.stack 100h
.data
    a dw 3
    b dw 1
    i dw -1
    k dw 0
    res dw ?

.code
mov ax, @data
mov ds, ax
mov dx, i ; i
shl dx, 1 ; 2i
add dx, i ; 3i
mov ax, a
cmp ax, b
jg func2

func1: ; a<=b
    mov cx, dx ; 3i
    neg cx ; -3i
    add cx, 12 ; 9-3(i-1)

    mov ax, cx ; 12-3i
    shl ax, 1 ; 24-6i
    sub ax, 18 ; -(6i-6)
    jmp func3

func2: ; a>b
    mov cx, dx ; 3i
    shl cx, 1 ; 6i
    add cx, 8 ; 6i+8
    neg cx ; -(6i+8)

    mov ax, cx ; -(6i+8)
```

```

        add ax, 28      ; 20-6i
        add ax, i       ; 20-5i
        add ax, i       ; 20-4i

func3:
        cmp k, 0
        jge max
        sub cx, ax      ; i2-i1

abs:      ; k<0
        neg cx
        js abs          ; |i1-i2|
        jmp result

max:      ; k>=0
        neg cx          ; -i2
        js max          ; |i2|
        cmp cx, 7
        jge result
        mov cx, 7
        jmp result

result:
        mov res, cx
        mov ah, 4ch
        int 21h
        end

```

**ПРИЛОЖЕНИЕ Б**  
**ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ**

Название файла: lab3.lst

```

dosseg
.model small
.stack 100h
.data

0000 0003          a dw 3
0002 0001          b dw 1
0004 FFFF          i dw -1
0006 0000          k dw 0
0008 0000          res dw ?

.code

0000 B8 ---- R      mov ax, @data
0003 8E D8          mov ds, ax
0005 8B 16 0004 R    mov dx, i ; i
0009 D1 E2          shl dx, 1 ; 2i
000B 03 16 0004 R    add dx, i ; 3i
000F A1 0000 R      mov ax, a
0012 3B 06 0002 R    cmp ax, b
0016 7F 11          jg func2

0018              func1: ; a<=b
0018 8B CA          mov cx, dx ; 3i
001A F7 D9          neg cx ; -3i
001C 83 C1 0C          add cx, 12 ; 9-3(i-1)

001F 8B C1          mov ax, cx ; 12-3i
0021 D1 E0          shl ax, 1 ; 24-6i
0023 2D 0012          sub ax, 18 ; -(6i-6)
0026 EB 17 90          jmp func3

0029              func2: ; a>b
0029 8B CA          mov cx, dx ; 3i
002B D1 E1          shl cx, 1 ; 6i
002D 83 C1 08          add cx, 8 ; 6i+8
0030 F7 D9          neg cx ; -(6i+8)

```

```

0032  8B C1                mov ax, cx    ; -(6i+8)
0034  05 001C              add ax, 28    ; 20-6i
0037  03 06 0004 R        add ax, i     ; 20-5i
003B  03 06 0004 R        add ax, i     ; 20-4i

003F                                func3:
003F  83 3E 0006 R 00      cmp k, 0
0044  7D 09                jge max
0046  2B C8                sub cx, ax    ; i2-i1

0048                                abs:    ; k<0
0048  F7 D9                neg cx
004A  78 FC                js abs        ; |i1-i2|
004C  EB 10 90              jmp result

004F                                max:    ; k>=0
004F  F7 D9                neg cx        ; -i2
0051  78 FC                js max        ; |i2|

```



Page 1-2

```
0053  83 F9 07                cmp cx, 7
0056  7D 06                jge result
0058  B9 0007            mov cx, 7
005B  EB 01 90                jmp result
```

```
005E                                result:
005E  89 0E 0008 R        mov res, cx
0062  B4 4C                mov ah, 4ch
0064  CD 21                int 21h
                                end
```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
DGROUP	. . . . .	GROUP		
_DATA	. . . . .	000A WORD PUBLIC		'DATA'
STACK	. . . . .	0100 PARA STACK		'STACK'
_TEXT	. . . . .	0066 WORD PUBLIC		'CODE'

Symbols:

	N a m e	Type	Value	Attr
A	. . . . .	L WORD	0000	_DATA
ABS	. . . . .	L NEAR	0048	_TEXT
B	. . . . .	L WORD	0002	_DATA
FUNC1	. . . . .	L NEAR	0018	_TEXT
FUNC2	. . . . .	L NEAR	0029	_TEXT
FUNC3	. . . . .	L NEAR	003F	_TEXT
I	. . . . .	L WORD	0004	_DATA
K	. . . . .	L WORD	0006	_DATA
MAX	. . . . .	L NEAR	004F	_TEXT
RES	. . . . .	L WORD	0008	_DATA
RESULT	. . . . .	L NEAR	005E	_TEXT
@CODE	. . . . .	TEXT	_TEXT	
@CODESIZE	. . . . .	TEXT	0	
@CPU	. . . . .	TEXT	0101h	

```
@DATASIZE . . . . . TEXT 0
@FILENAME . . . . . TEXT lab3
@VERSION . . . . . TEXT 510
```

```
64 Source Lines
```

```
64 Total Lines
```

```
28 Symbols
```

```
47990 + 459270 Bytes symbol space free
```

```
0 Warning Errors
```

```
0 Severe Errors
```