

**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.**

Студент гр. 1303

Насонов Я. К.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### Цель работы.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров вычисляет значения функций.

### 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  $b$ .

$$f4 = \begin{cases} / -(6*i - 4), & \text{при } a > b \\ \backslash 3*(i+2), & \text{при } a \leq b \end{cases}$$

$$f5 = \begin{cases} / 20 - 4*i, & \text{при } a > b \\ \backslash -(6*I - 6), & \text{при } a \leq b \end{cases}$$

$$f7 = \begin{cases} / |i1| + |i2|, & \text{при } k < 0 \\ \backslash \max(6, |i1|), & \text{при } k \geq 0 \end{cases}$$

### Выполнение работы

1. Из таблицы получен вариант набора функций, которые необходимо реализовать, приведенного в каталоге Задания.

2. Программа протранслирована с различными значениями переменных, результат выполнения набора функций зафиксирован в таблице;

Трансляция программы:

```
C:\>masm lab3.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lab3.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

    50122 + 461235 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

C:\>link LAB3.OBJ

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>
```

4. Программа выполнена в пошаговом режиме под управлением отладчика с фиксацией значений используемых переменных.

lab3.exe

№ теста	Тестируемый случай	Функции для данного случая	Данные	
			входные	выходные
1	$a > b$ $k < 0$	$f1 = 4 - 6 * i$ $f2 = 20 - 4 * i$ $f3 = \text{abs}(f1) + \text{abs}(f2)$	$a = 5, b = 2$ $k = -1$ $i = 3$	$f1 = -14 = \text{FFF2}$ $f2 = 8 = 0008$ $f3 = 22 = 0016$
2	$a > b$ $k \geq 0$	$f1 = 4 - 6 * i$ $f2 = 20 - 4 * i$ $f3 = \max(6, \text{abs}(f1))$	$a = 6, b = 4$ $k = 1$ $i = 2$	$f1 = -8 = \text{FFF8}$ $f2 = 12 = 000C$ $f3 = 8 = 0008$
3	$a \leq b$ $k < 0$	$f1 = 2 + 3 * i$ $f2 = 6 - 6 * i$ $f3 = \text{abs}(f1) + \text{abs}(f2)$	$a = 2, b = 5$ $k = -2$ $i = 1$	$f1 = 5 = 0005$ $f2 = 0 = 0000$ $f3 = 5 = 0005$
4	$a \leq b$ $k \geq 0$	$f1 = 2 + 3 * i$ $f2 = 6 - 6 * i$ $f3 = \max(6, \text{abs}(f1))$	$a = 3, b = 4$ $k = 2$ $i = 4$	$f1 = 14 = 000E$ $f2 = -18 = \text{FFEE}$ $f3 = 14 = 000E$

### Выводы

В ходе выполнения лабораторной работы были получены навыки разработки программы с заданными целочисленными значениями на языке программирования Ассемблер.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.asm

```
AStack SEGMENT STACK
    DW 2 DUP(?)
AStack ENDS

DATA SEGMENT
    a      DW 5
    b      DW 4
    i      DW 1
    k      DW 2
    i1     DW ?
    i2     DW ?
    res    DW ?
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
    Main PROC FAR

        push DS
        sub ax,ax
        push ax
        mov ax,DATA
        mov DS,ax

        ; f1: if a > b: i1 = 4 - 6 * i   else: i1 = 2 + 3 * i
        ; f2: if a > b: i2 = 20 - 4 * i  else: i2 = 6 - 6 * i

        mov ax, i
        shl ax, 1
        add ax, i ; ax = 3 * i

        mov bx, ax
        shl bx, 1 ; bx = 6 * i

        mov cx, a
        cmp cx, b

        jg a_greater_b

a_less_equal_b: ; a <= b
        ; f1 : i1 = 2 + 3 * i
        mov i1, 2
        add i1, ax

        ; f2 : i2 = 6 - 6 * i
        mov i2, 6
        sub i2, bx

        jmp f3

a_greater_b: ; a > b
        ; f1 : 4 - 6 * i
```

```

        mov i1, 4
        sub i1, bx

        ; f2 : 20 - 4 * i
        mov i2, 20
        add ax, i
        sub i2, ax

f3:
        mov ax, i1
        mov bx, i2

        cmp ax, 0
        jge cmp_k

pos_i1: ; i1 = |i1|
        neg ax

mov res, ax

cmp_k:
        mov cx, k
        cmp cx, 0
        jl neg_k ; k < 0

pos_k: ; k >= 0 : res = max(6, |i1|)
        cmp ax, 6
        jg final
        mov res, 6
        jmp final

neg_k: ; k < 0 : res = |i1| + |i2|
        cmp bx, 0
        jge pos_i2
        neg bx

pos_i2:
        add res, bx

final:
        ret
Main ENDP
CODE ENDS
END Main

```