

**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. 0383

Самара Р.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### Цель работы.

Разработать на языке Assembler программу, вычисляющую по заданным значениям параметров функции.

### 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 13

$$f2 = \begin{cases} / - (4*i+3), & \text{при } a > b \\ \backslash 6*i - 10, & \text{при } a \leq b \end{cases}$$

$$f8 = \begin{cases} / - (6*i+8), & \text{при } a > b \\ \backslash 9 - 3*(i-1), & \text{при } a \leq b \end{cases}$$

$$f3 = \begin{cases} / |i1 + i2|, & \text{при } k=0 \\ \backslash \min(i1,i2), & \text{при } k \neq 0 \end{cases}$$

### Выполнение работы.

Происходит расчет функций  $f_1$ ,  $f_2$ ,  $f_3$ . При организации ветвящихся процессов использовалась функция `str` и условные переходы. Для операций умножения использовался побитовый сдвиг влево и сложение.

Разработанный программный код см. в приложении А.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Проверка работы программы с помощью отладчика (все результаты заносились в реестр CX)

№ п/п	Входные данные	Полученное значение $i_1$	Полученное значение $i_2$	Полученное значение $res$	Комментарии
1	$a = 1$ $b = 2$ $i = 3$ $k = 4$	$0008h = 8$	$0003h = 3$	$0003h = 3$	Верно ( $a < b \Rightarrow$ $f_2 = 6 * 3 - 10 = 8$ ; $a < b \Rightarrow f_8 = 9 - 6 = 3$ ; $k \neq 0 \Rightarrow$ $\min(8, 3) = 3$ ).
2	$a = 1$ $b = 2$ $i = 3$ $k = 0$	$0008h = 8$	$0003h = 3$	$000Bh = 11$	Верно
3	$a = -3$ $b = -2$ $i = -1$ $k = 1$	$FFF0h = -16$	$000Fh = 15$	$FFF0h = -16$	Верно
4	$a = 2$ $b = 1$ $i = 3$ $k = 0$	$FFF1h = -15$	$FFE6h = -26$	$0029h = 41$	Верно

### Вывод.

Была разработана программа, вычисляющая функции в зависимости от параметров.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.asm

```
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
; Данные программы
DATA SEGMENT
; Директивы описания данных
a      DW 1
b      DW 2
i      DW 3
k      DW 0
i1     DW 0
i2     DW 0

DATA ENDS

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

; Головная процедура
Main PROC FAR
    push DS
    sub AX, AX
    push AX
    mov AX, DATA
    mov DS, AX
    mov CX, 0

    ; вычисление f2 и f8
    mov cx, i
    mov ax, cx
    mov bx, b
    cmp a, bx ; сравнение a и b
    jle fsecond ; a > b
    shl cx, 1
    shl cx, 1
    add cx, 3
    neg cx
    mov i1, cx
    sub cx, ax
    sub cx, ax
    add cx, -5
    mov i2, cx
    jmp finfun
fsecond: ; a <= b
    shl cx, 1
    add cx, ax
    neg cx
    add cx, 12
```

```

        mov i2, cx
        shl cx, 1
        neg cx
        add cx, 14
        mov i1, cx
        mov cx, i2

finfun:
        ;вычисление f3
        ;mov cx, i2
        mov bx, k
        cmp bx, 0
        je f3Second ; k != 0
        cmp cx, i1
        jle min1
        mov cx, i1          ; i2 <= i1
        jmp MainFinal
min1:
        jmp MainFinal

f3Second: ; k = 0
        add cx, i1
        cmp cx, 0
        jge MainFinal      ; |i1 + i2|
        neg cx
        jmp MainFinal

MainFinal: ; в cx лежит значение функции f3
        ret
Main      ENDP
CODE      ENDS
END Main

```

## Название файла: lab3.lst

01:32:00 Microsoft (R) Macro Assembler Version 5.10

12/2/21

Page

1-1

```

                                ; Стек программы
0000                                AStack SEGMENT  STACK
0000 000C[                                DW 12 DUP(?)
    ????                                ]

0018                                AStack ENDS
                                ;Данные программы
0000                                DATA            SEGMENT
                                ;Директивы описания данных
0000 0001                                a            DW 1
0002 0002                                b            DW 2
0004 0003                                i            DW 3
0006 0004                                k            DW 4
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 B9 0000          mov CX, 0

;вычисление f2 и f8
000C 8B 0E 0004 R      mov cx, i
0010 8B C1          mov ax, cx
0012 8B 1E 0002 R      mov bx, b
0016 39 1E 0000 R      cmp a, bx ; сравнение a и b
001A 7E 1B          jle fsecond ;a > b
001C D1 E1          shl cx, 1
001E D1 E1          shl cx, 1
0020 83 C1 03          add cx, 3
0023 F7 D9          neg cx
0025 89 0E 0008 R      mov i1, cx
0029 2B C8          sub cx, ax
002B 2B C8          sub cx, ax
002D 83 C1 FB          add cx, -5
0030 89 0E 000A R      mov i2, cx
0034 EB 1D 90          jmp finfun
0037          fsecond: ;a <= b
0037 D1 E1          shl cx, 1
0039 03 C8          add cx, ax
003B F7 D9          neg cx
003D 83 C1 0C          add cx, 12
0040 89 0E 000A R      mov i2, cx

```

Microsoft (R) Macro Assembler Version 5.10

12/2/21

01:32:00

Page

1-2

```

0044 D1 E1          shl cx, 1
0046 F7 D9          neg cx
0048 83 C1 0E          add cx, 14
004B 89 0E 0008 R      mov i1, cx
004F 8B 0E 000A R      mov cx, i2

0053          finfun:
;вычисление f3
;mov cx, i2
0053 8B 1E 0006 R      mov bx, k
0057 83 FB 00          cmp bx, 0
005A 74 10          je f3Second ; k != 0
005C 3B 0E 0008 R      cmp cx, i1
0060 7E 07          jle min1
0062 8B 0E 0008 R      mov cx, i1 ; i2 <= i1

```

```

0066 EB 12 90                                jmp MainFinal
0069                                min1:
0069 EB 0F 90                                jmp MainFinal

006C                                f3Second: ; k = 0
006C 03 0E 0008 R                          add cx, i1
0070 83 F9 00                              cmp cx, 0
0073 7D 05                                jge MainFinal ; |i1 + i2|
0075 F7 D9                                neg cx
0077 EB 01 90                                jmp MainFinal

007A                                MainFinal: ; в cx лежит значение фy
                                нкции f3
007A CB                                ret
007B                                Main ENDP
007B                                CODE ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10 12/2/21  
01:32:00

Symbols-1

#### Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK . . . . .		0018	PARA	STACK
CODE . . . . .		007B	PARA	NONE
DATA . . . . .		000C	PARA	NONE

#### Symbols:

	N a m e	Type	Value	Attr
A . . . . .		L WORD	0000	DATA
B . . . . .		L WORD	0002	DATA
F3SECOND . . . . .		L NEAR	006C	CODE
FINFUN . . . . .		L NEAR	0053	CODE
FSECOND . . . . .		L NEAR	0037	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
= 007B				
MAINFINAL . . . . .		L NEAR	007A	CODE
MIN1 . . . . .		L NEAR	0069	CODE
@CPU . . . . .		TEXT	0101h	
@FILENAME . . . . .		TEXT	LB3	
@VERSION . . . . .		TEXT	510	

82 Source Lines  
82 Total Lines  
20 Symbols

47992 + 461315 Bytes symbol space free

0 Warning Errors  
0 Severe Errors