

**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. 1303

Кузнецов Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить условные переходы и арифметические операции на ассемблере.

### **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  $b$ .

### **Ход работы.**

#### **Вариант 15**

Функции:

$f1:$	$/ 7 - 4*i$ , при $a > b$	$f2:$	$/ 20 - 4*i$ , при $a > b$
	$f3 = <$		$f5 = <$
	$\backslash 8 - 6*i$ , при $a \leq b$		$\backslash -(6*i - 6)$ , при $a \leq b$
$f3:$	$/  i1 + i2 $ , при $k = 0$		
	$f3 = <$		
	$\backslash \min(i1, i2)$ , при $k \neq 0$		

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 1;

3. Для минимизации кода программы в регистры  $ax$ ,  $bx$  заносятся в самом начале значения  $4i$ ,  $6i$  соответственно, чтобы не пересчитывать их несколько раз.

4. Также было выполнено раскрытие скобок  $f2 = -6i + 6$ , при  $a \leq b$

### Выполнение работы.

Результаты тестирования программы lab3.exe представлены в табл. 1.

Таблица 1 – Тестирование программы lab3.exe.

№ теста	Тестируемый случай	Функции для данного случая	Данные	
			входные	выходные
1	$a < b$ $k \neq 0$	$f1 = 8 - 6*i$ $f2 = -(6*i - 6)$ $f3 = \min(i1 + i2)$	$a = 4, b = 5$ $k = 1$ $i = 1$	$i1 = 2 = 0002$ $i2 = 0 = 0000$ $res = 0 = 0000$
2	$a > b$ $k = 0$	$f1 = 7 - 4*i$ $f2 = 20 - 4*i$ $f3 = \text{abs}(i1 + i2)$	$a = 5, b = 4$ $k = 0$ $i = 1$	$i1 = 3 = 0003$ $i2 = 16 = 0010$ $res = 19 = 0013$
3	$a = b$ $k = 0$	$f1 = 8 - 6*i$ $f2 = -(6*i - 6)$ $f3 = \text{abs}(i1 + i2)$	$a = 5, b = 5$ $k = 0$ $i = 2$	$i1 = -4 = \text{FFFC}$ $i2 = -6 = \text{FFFA}$ $res = 10 = 000A$
4	$a > b$ $k \neq 0$	$f1 = 7 - 4*i$ $f2 = 20 - 4*i$	$a = 13, b = -13$ $k = 13$	$i1 = -1 = \text{FFFF}$ $i2 = 12 = 000C$

		$f3 = \min(i1 + i2)$	$i = 2$	$res = -1 = FFFF$
--	--	----------------------	---------	-------------------

### **Вывод.**

Изучены условные переходы и арифметические операции на ассемблере.

## 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 0
    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 = 7 - 4*i else i1 = 8 - 6*i
    ; f2: if a>b i2 = 20 - 4*i i2 = else 6 - 6*i
    mov ax,i
    shl ax,1
    shl ax,1 ; ax = 4i
    mov bx,ax
    add bx,i
    add bx,i ; bx = 6i

    mov cx, a

    cmp cx, b
    jg f_case2 ; a > b
f_case1:
    ; f1
    mov i1,8
    sub i1,bx
    ; f2
    mov i2,6
    sub i2,bx
    jmp f_final
f_case2:
    ; f1
    mov i1,7
    sub i1,ax
    ; f2
```

```

        mov i2,20
        sub i2,ax
f_final:

        ; f3: if k = 0 res = |i1 + i2| else res = min(i1, i2)
        mov ax,i1
        mov bx,i2
        mov res,ax

        mov cx,k
        cmp k, 0
        je case2 ; k = 0
case1:
        cmp res,bx
        jbe final
        mov res,bx
        jmp final
case2:
        add res, bx
        cmp res, 0
        jge final ; res >= 0
        neg res
final:
        ret
Main ENDP
CODE ENDS
END Main

```