

**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.1381

\_\_\_\_\_

Дудко М.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### Цель работы.

Изучить представление и обработку целых чисел на языке Ассемблер.

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 № 10

$f2 = \begin{cases} / - (4*i+3) a > b \\ \backslash 6*i - 10 a \leq b \end{cases}$	$f5 = \begin{cases} / 20 - 4*i a > b \\ \backslash -(6*i-6) a \leq b \end{cases}$	$f8 = \begin{cases} /  i1 - i2  k < 0 \\ \backslash \max(7,  i2 ) k \geq 0 \end{cases}$
--	---	---

### Выполнение работы.

Созданы три сегмента AStack, DATA, CODE - сегмент стека, сегмент кода и сегмент данных соответственно. В сегменте данных объявлены переменные  $a$ ,  $b$ ,  $i$ ,  $k$ ,  $i1$ ,  $i2$ ,  $res$ . В сегменте кода находится процедура Main, в которой вычисляются значения данных в условии функций. Сначала в регистрах записывается значение  $bi$ . Далее с помощью функции  $cmp$  происходит сравнение значений переменных  $a$

и b. Команда jle проверяет условие  $a \leq b$ , при его выполнении производится переход по указанному адресу. При помощи команды jne (выполнение условия  $k! = 0$ ), производится переход по указанному адресу. Для совершения безусловных переходов в программе используется команда jmp. В переменную res записывается значение регистра ax, который содержит значение функции f3.

### Тестирование.

Результаты работы программы при  $a=-3$ ;  $b=-2$ ;  $i=22$ ;  $k=-2$  представлены в табл.1.

i1	i2	res	Правильность результата
007A (122)	FF82 (-126)	00F8 (256)	Верно

Таблица 1 – Результаты первого теста

Результаты работы программы при  $a=4$ ;  $b=2$ ;  $i=8$ ;  $k=0$  представлены в табл.2.

i1	i2	res	Правильность результата
FFDD (-35)	FFF4(-12)	000C (12)	Верно

Таблица 2 – Результаты второго теста

Результаты работы программы при  $a=2$ ;  $b=5$ ;  $i=5$ ;  $k=1$  представлены в табл.3.

i1	i2	res	Правильность результата
0014 (20)	FFE8 (-24)	0018 (24)	Верно

Таблица 3 – Результаты третьего теста

### Выводы.

В ходе выполнения лабораторной работы было изучено представление и обработка целых чисел, и организация ветвящихся процессов. Для выполнения задания была написана программа, которая вычисляет значения функций согласно заданным условиям.

# ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *lab3.asm*

```
AStack SEGMENT STACK
```

```
    DW 12 DUP(?)
```

```
AStack ENDS
```

```
DATA SEGMENT
```

```
    a DW 2
```

```
    b DW 5
```

```
    i DW 5
```

```
    k DW 1
```

```
    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
```

```
f12:
```

```
    mov ax,i
```

```
    add ax,i
```

```
    shl ax,1;a=4*i
```

```
    mov cx,a
```

```
    cmp cx,b
```

```
    jle
```

```
    f12step2;a<=b
```

```
f12step1:
```

```
    ;f1=-(4*i+3)
```

```
    mov i1,ax
```

```
    add i1,3
```

```
    neg i1
```

```
    mov dx,i1
```

```
    ;f2=20-4*i
```

```

        mov i2,20
        sub i2,ax
        mov dx,i2
        jmp f3
f12step2:
        ;f1=6*i-10
        sub ax,i
        shl ax,1
        mov i1,ax
        sub i1,10
        mov dx,i1
        ;f2=-(6*i-4)
        mov i2,ax
        sub i2,6
        neg i2
        mov dx,i2
f3:
        mov bx,i1
        sub bx,i2
        cmp bx,0
        jge f3step1
        ;i1>=0
        neg bx
f3step1:
        cmp i2,0
        jge f3step2
        ;i2>=0
        neg i2
f3step2:
        cmp k,0
        jge f3step3 ;
        k>=0
        mov ax,bx
        jmp f3end
f3step3:
        cmp i2,7
        jge max4
        mov ax,7
        jmp f3end
max4:

```

```
        mov ax,i2
f3end:
        mov res,ax
        ret
Main ENDP
CODE    ENDS
END Main
```

Название файла: *lab3.lst*

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

0018
                                AStack ENDS

0000
                                DATA SEGMENT

0000 0002
                                a DW 2

0002 0005
                                b DW 5

0004 0005
                                i DW 5

0006 0001
                                k DW 1

0008 0000
                                i1 DW ?

000A 0000
                                i2 DW ?

000C 0000
                                res DW ?

000E
                                DATA ENDS

0000
                                CODE SEGMENT

                                ASSUME CS:CODE,

DS:DATA, SS:AStack
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					f12:
0009	A1	0004	R		mov ax,i
000C	03	06 0004	R		add ax,i
0010	D1	E0			shl ax,1;a=4*i
0012	8B	0E 0000	R		mov cx,a
0016	3B	0E 0002	R		cmp cx,b
001A	7E	21			
jle f12step2;a<=b					
001C					f12step1:
					;f1=-(4*i+3)
001C	A3	0008	R		mov i1,ax
001F	83	06 0008	R 03		add i1,3
0024	F7	1E 0008	R		neg i1
0028	8B	16 0008	R		mov dx,i1
					;f2=20-4*i
002C	C7	06 000A	R 0014		mov i2,20
0032	29	06 000A	R		sub i2,ax
0036	8B	16 000A	R		mov dx,i2
003A	EB	23 90			jmp f3
003D					f12step2:
					;f1=6*i-10
003D	2B	06 0004	R		



0041	D1 E0	sub ax,i
0043	A3 0008 R	shl ax,1
0046	83 2E 0008 R 0A	mov i1,ax
004B	8B 16 0008 R	sub i1,10
		mov dx,i1
		;f2=-(6*i-
4)		
004F	A3 000A R	mov i2,ax
0052	83 2E 000A R 06	sub i2,6
0057	F7 1E 000A R	neg i2
005B	8B 16 000A R	mov dx,i2
005F		f3:

005F	8B 1E 0008 R	
		mov bx,i1
0063	2B 1E 000A R	
		sub bx,i2
0067	83 FB 00	
		cmp bx,0
006A	7D 02	
		jge f3step1
;i1>=0		
006C	F7 DB	
		neg bx
006E		
		f3step1:
006E	83 3E 000A R 00	
		cmp i2,0
0073	7D 04	
		jge f3step2
;i2>=0		
0075	F7 1E 000A R	
		neg i2
0079		
		f3step2:
0079	83 3E 0006 R 00	
		cmp k,0
007E	7D 05	
		jge f3step3 ;
k>=0		
0080	8B C3	
		mov ax,bx
0082	EB 11 90	
		jmp f3end
0085		
		f3step3:
0085	83 3E 000A R 07	
		cmp i2,7
008A	7D 06	
		jge max4
008C	B8 0007	

```

008F  EB 04 90

0092

0092  A1 000A R

0095

0095  A3 000C R

0098  CB

0099

0099

```

```

mov ax,7

jmp f3end

max4:

mov ax,i2

f3end:

mov res,ax

ret

Main      ENDP

CODE      ENDS

```

```

Microsoft (R) Macro Assembler Version 5.10
13:11:32

```

```

END Main
11/4/22

Symbols-1

```

# Segments and Groups:

N a m e		Length	Align	Combine	Class
ASTACK	. . . . .	0018	PARA		
			STACK		
CODE	. . . . .	0099	PARA		
			NONE		
DATA	. . . . .	000E	PARA		
			NONE		

# Symbols:

N a m e		Type	Value
Attr			
A	. . . . .	L WORD	0000

	DATA
B . . . . .	L WORD 0002 DATA
F12 . . . . .	L NEAR 0009 CODE
F12STEP1 . . . . .	L NEAR 001C CODE
F12STEP2 . . . . .	L NEAR 003D CODE
F3 . . . . .	L NEAR 005F CODE
F3END . . . . .	L NEAR 0095 CODE
F3STEP1 . . . . .	L NEAR 006E CODE
F3STEP2 . . . . .	L NEAR 0079 CODE
F3STEP3 . . . . .	L NEAR 0085 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 CODELength =
0099	
MAX4 . . . . .	L NEAR 0092 CODE
RES . . . . .	L WORD 000C DATA
@CPU . . . . .	TEXT 0101h
@FILENAME . . . . .	TEXT LAB3

@VERSION . . . . .

TEXT 510

78 Source Lines

78 Total Lines

25 Symbols

48056 + 461251 Bytes symbol space free

0 Warning Errors

0 Severe Errors