

**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

\_\_\_\_\_

Куликов А. В.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2021

### **Цель работы.**

Изучить представление и обработку целых чисел, организацию ветвящихся процессов на языке Ассемблера.

### **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 функции  $f1$ ,  $f2$ ,  $f3$ :

$/ - (4*i+3)$ , при  $a > b$

$f1 = <$

$\backslash 6*i - 10$ , при  $a \leq b$

$/ - (6*i+8)$ , при  $a > b$

$f2 = <$

$\backslash 9 - 3*(i-1)$ , при  $a \leq b$

$/ |i1 + i2|$ , при  $k=0$

$f3 = <$

$\backslash \min(i1, i2)$ , при  $k \neq 0$

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  $b$ .

### **Выполнение работы.**

В ходе выполнения работы были использованы инструкции `cmp`, `jle`, `jge`, `jl`, `jg`, `jne`, `je`, использованные для реализации ветвления и сравнения веденных чисел.

Все возможные варианты работы кода программы приведены в Табл.1.

Табл.1

Значения a, b, i, k	Результат вычисления i1	Результат вычисления i2	Результат вычисления res	Прим.
a = 1 b = 0 i = -5 k = 0	11 (hex) = 17 (dec)	16 (hex) = 22 (dec)	27 (hex) = 39 (dec)	Верно
a = 1 b = 0 i = -5 k = 1	11 (hex) = 17 (dec)	16 (hex) = 22 (dec)	11 (hex) = 17 (dec)	Верно
a = -1 b = 10 i = 2 k = 0	2 (hex) = 2 (dec)	6 (hex) = 6 (dec)	8 (hex) = 8 (dec)	Верно
a = -1 b = -1 i = -2 k = -7	FFEA(hex) = -22 (dec)	12 (hex) = 18 (dec)	FFEA (hex) = -22 (dec)	Верно

### Выводы.

В ходе лабораторной работы были изучены представление и обработка целых чисел и организация ветвящихся процессов в языке Ассемблера.

## ПРИЛОЖЕНИЕ А

### Тексты исходных файлов программ

#### lab3.asm

AStack SEGMENT STACK

DW 12 DUP(?)

AStack ENDS

DATA SEGMENT

a DW 0

b DW 0

i DW 0

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 DX, AX

mov a, 0

mov b, 0

mov i, 0

mov k, 0

$;bx = 3 * i$

mov bx, i

shl bx, 1

add bx, i

mov cx, a

cmp cx, b

jle L1

$;a > b$

$;i1 = -(4 * i + 3)$

add bx, i

add bx, 3

neg cx

mov i1, bx

$;i2 = -(6 * i + 8)$

add bx, 4

shl bx, 1

neg bx

mov i2, bx

jmp L2

;a <= b

L1:

;i2 = 9 - 3 \* (i - 1)

neg bx

add bx, 12

mov i2, bx

;i1 = 6 \* i - 10

neg bx

shl bx, 1

add bx, 2

mov i1, bx

L2:

cmp k, 0

jne L3

;k == 0

mov AX, i1

add AX, i2

cmp AX, 0

jg endmain

neg AX

jmp endmain

;k != 0

L3:

```
mov Ax, i1  
cmp AX, i2  
jle endmain
```

```
mov AX, i2
```

```
endmain:  
ret
```

```
Main ENDP
```

```
CODE ENDS
```

```
END Main
```

## ПРИЛОЖЕНИЕ Б

### Тексты файлов диагностических сообщений программ

lab3.lst MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10

12/8/21 00:57:02

PAGE 1-1

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

0018          ASTACK ENDS

0000          DATA SEGMENT
0000 0000          A DW 0
0002 0000          B DW 0
0004 0000          I DW 0
0006 0000          K DW 0
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 8B D0          MOV DX, AX

0009 C7 06 0000 R 0000          MOV A, 0
000F C7 06 0002 R 0000          MOV B, 0
0015 C7 06 0004 R 0000          MOV I, 0
001B C7 06 0006 R 0000          MOV K, 0
```

;BX = 3 \* I



0021	8B 1E 0004 R	MOV BX, I
0025	D1 E3	SHL BX, 1
0027	03 1E 0004 R	ADD BX, I

002B	8B 0E 0000 R	MOV CX, A
002F	3B 0E 0002 R	CMP CX, B
0033	7E 1B	JLE L1

```

;A > B
;I1 = -(4 * I + 3)

```

0035	03 1E 0004 R	ADD BX, I
0039	83 C3 03	ADD BX, 3
003C	F7 D9	NEG CX
003E	89 1E 0008 R	MOV I1, BX

```

;I2 = -(6 * I + 8)

```

0042	83 C3 04	ADD BX, 4
0045	D1 E3	SHL BX, 1
0047	F7 DB	NEG BX

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10  
00:57:02

12/8/21

PAGE 1-2

0049	89 1E 000A R	MOV I2, BX
004D	EB 15 90	JMP L2

0050		;A <= B
		L1:

		;I2 = 9 - 3 * (I - 1)
0050	F7 DB	NEG BX
0052	83 C3 0C	ADD BX, 12
0055	89 1E 000A R	MOV I2, BX

		;I1 = 6 * I - 10
0059	F7 DB	NEG BX
005B	D1 E3	SHL BX, 1
005D	83 C3 02	ADD BX, 2
0060	89 1E 0008 R	MOV I1, BX

0064		L2:
------	--	-----

```

0064 83 3E 0006 R 00      CMP K, 0
0069 75 11                JNE L3

                        ;K == 0

006B A1 0008 R            MOV AX, I1
006E 03 06 000A R        ADD AX, I2
0072 3D 0000            CMP AX, 0
0075 7F 11              JG ENDMAN
0077 F7 D8              NEG AX
0079 EB 0D 90            JMP ENDMAN

```

```

                        ;K != 0
007C                      L3:
007C A1 0008 R            MOV AX, I1
007F 3B 06 000A R        CMP AX, I2
0083 7E 03              JLE ENDMAN

```

```

0085 A1 000A R            MOV AX, I2

```

```

0088                      ENDMAN:
0088 CB                  RET
0089                      MAIN ENDP
0089                      CODE ENDS

```

END MAIN

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10  
00:57:02

12/8/21

## SYMBOLS-1

### SEGMENTS AND GROUPS:

N A M E	LENGTH	ALIGN	COMBINE CLASS
ASTACK .....	0018	PARA	STACK
CODE .....	0089	PARA	NONE
DATA .....	000C	PARA	NONE

### SYMBOLS:

N A M E	TYPE	VALUE	ATTR
A .....	L WORD	0000	DATA

```

B ..... L WORD  0002 DATA

ENDMAIN ..... L NEAR  0088 CODE

I ..... L WORD  0004 DATA
I1 ..... L WORD  0008 DATA
I2 ..... L WORD  000A DATA

K ..... L WORD  0006 DATA

L1 ..... L NEAR  0050 CODE
L2 ..... L NEAR  0064 CODE
L3 ..... L NEAR  007C CODE

MAIN ..... F PROC  0000 CODE      LENGTH = 0089

@CPU ..... TEXT 0101H
@FILENAME ..... TEXT LAB3
@VERSION ..... TEXT 510

```

```

97 SOURCE LINES
97 TOTAL LINES
19 SYMBOLS

```

48058 + 461249 BYTES SYMBOL SPACE FREE

```

0 WARNING ERRORS
0 SEVERE ERRORS

```