

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Представление и обработка целых чисел. Организация**  
**ветвящихся процессов**

Студент гр. 9383

\_\_\_\_\_

Арутюнян С.Н.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

## Содержание

1. Цель работы.....	3
2. Текст программы lab3.asm.....	5
3. Текст листинга lab3.lst.....	8
Выводы.....	14

## 1. Цель работы

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  $b$ .

Вид функций:

$f1 = 15 - 2i$  при  $a > b$ ;  $3i + 4$  при  $a \leq b$

$f2 = -(4i + 3)$  при  $a > b$ ;  $6i - 10$  при  $a \leq b$

$f3 = \min(i1, i2)$  при  $k = 0$ ;  $\max(i1, i2)$  при  $k \neq 0$

## 2. Текст программы lab3.asm

; Стек программы

AStack SEGMENT STACK

DW 12 DUP(?)

AStack ENDS

; Данные программы

DATA SEGMENT

a DW 1

b DW 2

i DW 3

k DW 4

i1 DW ?

i2 DW ?

result DW ?

DATA ENDS

; Код программы

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура

Main PROC FAR

mov ax, DATA

mov ds, ax

mov ax, a

mov bx, b

cmp ax, bx

jg great\_branch ; if ax > bx

less\_branch:

; f1 (3i + 4) =====

; 3\*i

mov dx, i

; \*2

shl dx, 1

; +i

add dx, i

; +4

add dx, 4

mov i1, dx

; =====

; f2 (6i - 10) =====

; dx = 3i + 4

shl dx, 1

; dx = 6i + 8

add dx, -18

mov i2, dx

; =====

jmp res

great\_branch:

; f1 (15 - 2i) =====

; 2\*i

mov dx, i

shl dx, 1

; -15

sub dx, 15

neg dx

mov i1, dx

; =====

; f2 (-(4i + 3)) =====

neg dx

; dx = -15 + 2i

shl dx, 1

; dx = -30 + 4i

add dx, 33

neg dx

mov i2, dx

; =====

; f3

res:

mov ax, i1

mov bx, i2

mov cx, k

; упорядочиваем ax и bx

```
    cmp ax, bx
    jg min
    jmp res2
```

```
min:
    xchg ax, bx
```

```
res2:
    cmp k, 0
    jz res_zero
```

```
; k != 0
res_nonzero:
    ; максимальный в bx
    mov result, bx
    jmp end_prog
```

```
; k = 0
res_zero:
    ; минимальный в ax
    mov result, ax
```

```
end_prog:
    mov ah, 4ch
    int 21h
```

```
Main ENDP
CODE ENDS
END Main
```

### 3. Текст листинга lab3.lst

#Microsoft (R) Macro Assembler Version 5.10

11/5/20 00:30:09

Page 1-1

; Стек программы

0000 AStack SEGMENT STACK

0000 000C [DW 12 DUP(?) ????]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

0000 0001 a DW 1

0002 0002 b DW 2

0004 0003 i DW 3

0006 0004 k DW 4

0008 0000 i1 DW ?

000A 0000 i2 DW ?

000C 0000 result DW ?

000E DATA ENDS

; Код программы

0000 CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура

0000 Main PROC FAR

0000 B8 ---- R mov ax, DATA

0003 8E D8 mov ds, ax



```

0005 A1 0000 R      mov ax, a
0008 8B 1E 0002 R      mov bx, b
000C 3B C3          cmp ax, bx

000E 7F 1D          jg great_branch ; if ax > bx

```

```

0010          less_branch:
                ; f1 (3i + 4) =====
                ; 3*i

0010 8B 16 0004 R      mov dx, i
                ; *2

0014 D1 E2          shl dx, 1
                ; +i

0016 03 16 0004 R      add dx, i
                ; +4

001A 83 C2 04          add dx, 4

001D 89 16 0008 R      mov i1, dx
                ; =====

                ; f2 (6i - 10) =====
                ; dx = 3i + 4

0021 D1 E2          shl dx, 1
                ; dx = 6i + 8

0023 83 C2 EE          add dx, -18

```

#Microsoft (R) Macro Assembler Version 5.10

11/5/20 00:30:09

Page 1-2

```

0026 89 16 000A R      mov i2, dx

```

```

; =====

002A EB 1D 90          jmp res

002D          great_branch:
; f1 (15 - 2i) =====
; 2*i

002D 8B 16 0004 R      mov dx, i
0031 D1 E2            shl dx, 1
; -15

0033 83 EA 0F          sub dx, 15
0036 F7 DA            neg dx

0038 89 16 0008 R      mov i1, dx
; =====

; f2 (-(4i + 3)) =====

003C F7 DA            neg dx
; dx = -15 + 2i

003E D1 E2            shl dx, 1
; dx = -30 + 4i

0040 83 C2 21          add dx, 33
0043 F7 DA            neg dx

0045 89 16 000A R      mov i2, dx
; =====

; f3

```

```

0049                                res:
0049 A1 0008 R                        mov ax, i1
004C 8B 1E 000A R                    mov bx, i2
0050 8B 0E 0006 R                    mov cx, k

                                ; упорядочиваем ax и bx
0054 3B C3                          cmp ax, bx
0056 7F 03                          jg min
0058 EB 02 90                        jmp res2

```

```

005B                                min:
005B 93                              xchg ax, bx

```

```

005C                                res2:
005C 83 3E 0006 R 00                 cmp k, 0
0061 74 07                          jz res_zero

```

```

                                ; k != 0
0063                                res_nonzero:
                                ; максимальный в bx
0063 89 1E 000C R                    mov result, bx
0067 EB 04 90                        jmp end_prog

```

#Microsoft (R) Macro Assembler Version 5.10

11/5/20 00:30:09

Page 1-3

```

                                ; k = 0
006A                                res_zero:

```

```

; минимальный в ax
006A A3 000C R      mov result, ax

```

```

006D                end_prog:
006D B4 4C          mov ah, 4ch
006F CD 21          int 21h

```

```

0071                Main ENDP
0071                CODE ENDS
                END Main

```

```
#Microsoft (R) Macro Assembler Version 5.10      11/5/20 00:30:09
```

# Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK .....	0018	PARA		STACK
CODE .....	0071	PARA		NONE
DATA .....	000E	PARA		NONE

## Symbols:

N a m e	Type	Value	Attr
A .....	L WORD	0000	DATA
B .....	L WORD	0002	DATA
END_PROG .....	L NEAR	006D	CODE
GREAT_BRANCH .....	L NEAR	002D	CODE

```

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

K ..... L WORD  0006 DATA

LESS_BRANCH ..... L NEAR  0010 CODE

MAIN ..... F PROC  0000 CODE      Length = 0071
MIN ..... L NEAR  005B CODE

RES ..... L NEAR  0049 CODE
RES2 ..... L NEAR  005C CODE
RESULT ..... L WORD  000C DATA
RES_NONZERO ..... L NEAR  0063 CODE
RES_ZERO ..... L NEAR  006A CODE

@CPU ..... TEXT 0101h
@FILENAME ..... TEXT lab2
@VERSION ..... TEXT 510

117 Source Lines
117 Total Lines
24 Symbols

47962 + 459298 Bytes symbol space free

0 Warning Errors
0 Severe Errors

```

## **Выводы**

В процессе выполнения лабораторной работы было изучено представление и обработка целых чисел, а также получены навыки работы с организацией ветвящихся процессов.