

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. 1383

Федорова О.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Научиться обрабатывать целые числа, создавать ветвления 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 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 1.

$$/ 15-2*i, a>b$$

$$f1 = <$$

$$\backslash 3*i+4, a\leq b$$

$$/ - (4*i+3), a>b$$

$$f2 = <$$

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

$$/ \min(i1,i2), k=0$$

$$f3 = <$$

$$\backslash \max(i1,i2), k\neq 0$$

Таблица 1. Примеры

Номер	Введенные данные	Вывод
1.	$a = -3$	$I1 = 7 = 0007$
	$b = 2$	$i2 = -4 = \text{FFFC}$
	$i = 1$	$\text{res} = -4 = \text{FFFC}$
	$k = 0$	
2.	$a = 2$	$I1 = 17 = 0011$
	$b = -2$	$i2 = 1 = 0001$
	$i = -1$	$\text{res} = 17 = 0011$
	$k = 1$	
3.	$a = 2$	$I1 = -5 = \text{FFFB}$
	$b = 2$	$i2 = -28 = \text{FFE4}$
	$i = -3$	$\text{res} = -5 = \text{FFFB}$
	$k = 1$	

4.	a = 2 b = 2 i = 2	I1 = 10 = 000A i2 = 2 = 0002 res = 10 = 000A
5.	k = 1 a = 2 b = 2 i = 2	I1 = 10 = 000A i2 = 2 = 0002 res = 2 = 0002
6.	k = 0 a = 2 b = 2 i = -3	I1 = -5 = FFFB i2 = -28 = FFE4 res = -28 = FFE4
7.	k = 0 a = 0 b = 1 i = 3	I1 = 13 = 000D i2 = 8 = 0008 res = 8 = 0008
8.	k = 0 a = 2 b = 1 i = 1	I1 = 14 = 000D i2 = -7 = FFF9 res = 14 = 000D
9.	k = 1 a = 2 b = 1 i = 1	I1 = 14 = 000D i2 = -7 = FFF9 res = -7 = FFF9
10.	k = 0 a = 2 b = 1 i = -1 k = 0	I1 = 17 = 0011 i2 = 1 = 0001 res = 1 = 0001

Примечания:

Для меньшего количества кода использовался побитовый сдвиг влево(умножение на 2, shl cx,1)

Вывод:

Были изучены арифметические операции над целыми числами на языке ассемблер.

\

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
lbb3.asm
ind EQU 2
n1 EQU 500
n2 EQU -50
AStack SEGMENT STACK
DW 12 DUP(?)
AStack ENDS
DATA SEGMENT
a DW 2
b DW 1
i DW -1
k DW 0
i1 DW ?
i2 DW ?
res DW ?
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

    ;/ 15-2*i , DžŃ ħ Dž a>b
;f1 = <
    ;\ 3*i+4 , DžŃ ħ Dž a<=b

    ;/ - (4*i+3) , DžŃ ħ Dž a>b
;f2 = <
    ;\ 6*i -10 , DžŃ ħ Dž a<=b
```

```

    ;/ min(i1,i2), ÐžÑ ħ Ðž k=0
;f3 = <
    ;\ max(i1,i2), ÐžÑ ħ Ðž k/=0

```

Main PROC FAR

```

    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
    mov cx,a
f1:
    cmp cx,b
    jle f1_b
f1_a:
    mov ax, i
    ;add ax, i
    shl ax,1
    mov dx,ax
    mov ax, 15
    sub ax, dx
    mov i1, ax
    push i1
    jmp f2_a
f1_b:
    mov ax, i
    shl ax,1
    add ax, i

```

```

;add ax, i
;add ax, i
add ax, 4
mov i1, ax
push i1
jmp f2_b
f2_a:
mov ax, i
add ax, i
;add ax, i
;add ax, i
shl ax, 1
add ax, 3
mov dx, ax
mov ax, 0
sub ax, dx
mov i2, ax
push i2
jmp f3
f2_b:
mov ax, i
;add ax, i
;add ax, i
;add ax, i
;add ax, i
;add ax, i
;add ax, i
shl ax, 1
add ax, i
shl ax, 1

```



```

    sub ax, 10
    mov i2, ax
    push i2
f3:
    pop ax;i2
    pop dx;i1
    mov cx, k
    cmp cx,0
    jcxz f3_null
    cmp ax,dx
    jl res_sec
    mov res, ax
    jmp print
f3_null:
    cmp ax,dx
    jg res_sec
    mov res, ax
    jmp print
res_sec:
    mov res, dx
print:
    push res
    pop dx
    add dl,10h
    int 21h
    ret
Main ENDP
CODE ENDS
END Main

```

ПРИЛОЖЕНИЕ В
ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

#Microsoft (R) Macro Assembler Version 5.10

11/13/22 17:48:3

Page 1-1

ind EQU 2

LAB3.ASM(1): warning A4001: Extra characters on line

= 01F4 n1 EQU 500

=-0032 n2 EQU -50

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP(?)

????

]

0018 AStack ENDS

0000 DATA SEGMENT

0000 0002 a DW 2

0002 0001 b DW 1

0004 FFFF i DW -1

0006 0000 k DW 0

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

;/ 15-2*i , ĐžŃ ħ Đž a>b

```

;f1 = <
        ;\ 3*i+4 , DžŃ ħ Dž a<=b

        ;/ - (4*i+3) , DžŃ ħ Dž a>b
;f2 = <
        ;\ 6*i -10 , DžŃ ħ Dž a<=b

        ;/ min(i1,i2), DžŃ ħ Dž k=0
;f3 = <
        ;\ max(i1,i2), DžŃ ħ Dž k/=0

```

```

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 8B 0E 0000 R  mov cx,a
000D          f1:
000D 3B 0E 0002 R  cmp cx,b
0011 7E 16          jle f1_b
0013          f1_a:
0013 A1 0004 R    mov ax, i
                ;add ax, i

0016 D1 E0          shl ax,1
0018 8B D0          mov dx,ax
001A B8 000F      mov ax, 15

```

```

001D 2B C2                sub ax, dx
001F A3 0008 R            mov il, ax
0022 FF 36 0008 R        push il

```

#Microsoft (R) Macro Assembler Version 5.10

11/13/22 17:48:3

Page 1-2

```

0026 EB 17 90            jmp f2_a
0029                    fl_b:
0029 A1 0004 R            mov ax, i
002C D1 E0                shl ax, 1
002E 03 06 0004 R        add ax, i
                        ;add ax, i
                        ;add ax, i
0032 05 0004            add ax, 4
0035 A3 0008 R            mov il, ax
0038 FF 36 0008 R        push il
003C EB 1E 90            jmp f2_b
003F                    f2_a:
003F A1 0004 R            mov ax, i
0042 03 06 0004 R        add ax, i
                        ;add ax, i
                        ;add ax,i
0046 D1 E0                shl ax, 1
0048 05 0003            add ax, 3
004B 8B D0                mov dx,ax
004D B8 0000            mov ax, 0
0050 2B C2                sub ax, dx

```

0052	A3 000A R	mov i2, ax
0055	FF 36 000A R	push i2
0059	EB 16 90	jmp f3
005C		f2_b:
005C	A1 0004 R	mov ax, i
		;add ax, i
		;add ax, i
		;add ax, i
		;add ax, i
		;add ax, i
005F	D1 E0	shl ax, 1
0061	03 06 0004 R	add ax, i
0065	D1 E0	shl ax, 1
0067	2D 000A	sub ax, 10
006A	A3 000A R	mov i2, ax
006D	FF 36 000A R	push i2
0071		f3:
0071	58	pop ax;i2
0072	5A	pop dx;i1
0073	8B 0E 0006 R	mov cx, k
0077	83 F9 00	cmp cx,0
007A	E3 0A	jcxz f3_null
007C	3B C2	cmp ax,dx
007E	7C 10	jl res_sec
0080	A3 000C R	mov res, ax
0083	EB 0F 90	jmp print
0086		f3_null:
0086	3B C2	cmp ax,dx
0088	7F 06	jg res_sec

```

008A A3 000C R      mov res, ax
008D EB 05 90      jmp print
0090                res_sec:
0090 89 16 000C R      mov res, dx

```

#Microsoft (R) Macro Assembler Version 5.10

11/13/22 17:48:3

Page 1-3

```

0094                print:
0094 FF 36 000C R      push res
0098 5A                pop dx
0099 80 C2 10          add dl,10h
009C CD 21            int 21h
009E CB                ret
009F                Main ENDP
009F                CODE ENDS
                END Main

```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA		STACK
CODE	009F	PARA		NONE
DATA	000E	PARA		NONE

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F1	L NEAR	000D	CODE
F1_A	L NEAR	0013	CODE
F1_B	L NEAR	0029	CODE
F2_A	L NEAR	003F	CODE
F2_B	L NEAR	005C	CODE
F3	L NEAR	0071	CODE
F3_NULL	L NEAR	0086	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 = 009F

N1 NUMBER 01F4

N2 NUMBER -0032

PRINT L NEAR 0094 CODE

RES L WORD 000C DATA

RES_SEC L NEAR 0090 CODE

@CPU TEXT 0101h

@FILENAME TEXT LAB3

@VERSION TEXT 510

#Microsoft (R) Macro Assembler Version 5.10

11/13/22 17:48:3

Symbols-2

114 Source Lines

114 Total Lines

27 Symbols

48008 + 459252 Bytes symbol space free

1 Warning Errors

0 Severe Errors