

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

Сапожников А.Э.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Получение знаний о ветвлениях и работе с целочисленными значениями 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 .

Выполнение работы.

Вариант 3.7.5

$i1=f1(a,b,i) := a>b ? : 7 - 4*i : 8 - 6*i$

$i2=f2(a,b,i) := a>b ? -(4*i-5) : 10 - 3*i$

$res=f3(i1,i2,k) := \min(|i1|, 6) ? k=0 : |i1|+|i2|$

1. В целях оптимизации сразу обрабатываем обе функции:

($f1$ и $f3$ для ситуаций $a > b$ или $a \leq b$)

2. формулы:

$7 - 4*i = 7 + (-i < 2)$

$8 - 6*i = 8 - (i < 2 + i+i)$

$$-(4*i-5) = 5 - (i < 2)$$

$$10 - 3*i = 10 + (-i < 2) + i$$

для упрощения расчетов считается $4*i$

Таблица 1- тесты программы

Номер	Входные данные	Результат
1	a = 0 b = 0 i = 0 k = 0	i1 = 8-0 i2 = 10 res = 6
2	a = 0 b = -1 i = 1 k = 0	i1 = 3 i2 = 1 res = 1
3	a = 0 b = -1 i = -1 k = 0	i1 = 11 i2 = 9 res = 6
4	a = 1 b = 1 i = 1 k = 1	i1 = 2 i2 = 7 res = 9
5	a = 2 b = 3 i = 4 k = -1	i1 = -16 i2 = -2 res = 18
6	a = 3 b = 2 i = -2 k = 1	i1 = 15 i2 = 13 res = 28
7	a = -3 b = 2 i = 2 k = 0	i1 = -4 i2 = 4 res = 4

Выводы.

Мною были изучены работа с целыми числами на языке Ассемблера и ветвления.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr3.asm

Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1

Page 1-1

```

=-0003                a EQU -3
= 0002                b EQU 2
= 0002                i EQU 2
= 0001                k EQU 1
0000                  AStack SEGMENT STACK
0000 000C[            DW 12 DUP(?)
    ????
    ]

0018                  AStack ENDS

0000                  DATA SEGMENT

0000 0000              res DW 0
0002                  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 0002           mov ax, i
0007 D1 E0             shl ax, 1
0009 D1 E0             shl ax, 1
000B 8B D8             mov bx, ax
```

```

000D  B9 FFFD                mov cx, a
0010  83 F9 02                cmp cx, b
0013  7E 0D                    jle fless

0015  F7 D8                    neg ax
0017  05 0007                add ax, 7
001A  50                      push ax ;i1

001B  05 FFFE                add ax, -2
001E  50                      push ax ;i2
001F  EB 16 90                jmp f3
0022                                fless:
0022  05 0002                add ax, i
0025  05 0002                add ax, i
0028  F7 D8                    neg ax
002A  05 0008                add ax, 8
002D  50                      push ax

002E  05 FFF8                add ax, -8
0031  D1 F8                    sar ax, 1
0033  05 000A                add ax, 10
0036  50                      push ax

0037                                f3:
0037  5A                      pop dx ;i2
Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1

```

Page 1-2

```

0038  59                      pop cx ;i1
0039  83 F9 00                cmp cx, 0
003C  79 02                    jns ilabs ;|i2| = |i2|
003E  F7 D9                    neg cx
0040                                ilabs:
0040  B8 0001                mov ax, k
0043  3D 0000                cmp ax, 0
0046  75 0B                    jne f3v2
0048  83 F9 06                cmp cx, 6
004B  7C 0F                    jl finish
004D  B9 0006                mov cx, 6
0050  EB 0A 90                jmp finish

```

```

0053          f3v2:
0053  83 FA 00          cmp dx, 0
0056  79 02          jns i2abs ; |i1| =
|i1|
0058  F7 DA          neg dx
005A          i2abs:
005A  03 CA          add cx, dx

005C          finish:
005C  51          push cx
005D  59          pop cx
005E  CB          ret
005F          Main ENDP
005F          CODE ENDS
          END Main

```

Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1

Symbols-1

Segments and Groups:

	N a m e	Length	Align
Combine Class			
STACK	ASTACK	0018	PARA
NONE	CODE	005F	PARA
NONE	DATA	0002	PARA

Symbols:

Attr	N a m e	Type	Value
-0003	A		NUMBER
0002	B		NUMBER

	F3	L	NEAR
0037	CODE		
	F3V2	L	NEAR
0053	CODE		
	FINISH	L	NEAR
005C	CODE		
	FLESS	L	NEAR
0022	CODE		
	I		NUMBER
0002			
	I1ABS	L	NEAR
0040	CODE		
	I2ABS	L	NEAR
005A	CODE		
	K		NUMBER
0001			
	MAIN	F	PROC
0000	CODE Length = 005F		
	RES	L	WORD
0000	DATA		
	@CPU	TEXT	0101h
	@FILENAME	TEXT	LR3
	@VERSION	TEXT	510

78 Source Lines
 78 Total Lines
 20 Symbols

47978 + 461329 Bytes symbol space free

0 Warning Errors
 0 Severe Errors

ПРИЛОЖЕНИЕ В

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: LR3.lst

Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1

Page 1-1

```
=-0003          a EQU -3
= 0002          b EQU 2
= 0002          i EQU 2
= 0001          k EQU 1
0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
    ????
    ]

0018          AStack ENDS

0000          DATA SEGMENT

0000 0000          res DW 0
0002          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 0002          mov ax, i
0007 D1 E0          shl ax, 1
0009 D1 E0          shl ax, 1
000B 8B D8          mov bx, ax

000D B9 FFFD          mov cx, a
```

```

0010  83 F9 02                cmp cx, b
0013  7E 0D                    jle fless

0015  F7 D8                    neg ax
0017  05 0007                  add ax, 7
001A  50                        push ax ;i1

001B  05 FFFE                    add ax, -2
001E  50                        push ax ;i2
001F  EB 16 90                  jmp f3
0022                                fless:
0022  05 0002                  add ax, i
0025  05 0002                  add ax, i
0028  F7 D8                    neg ax
002A  05 0008                  add ax, 8
002D  50                        push ax

002E  05 FFF8                    add ax, -8
0031  D1 F8                    sar ax, 1
0033  05 000A                  add ax, 10
0036  50                        push ax

0037                                f3:
0037  5A                        pop dx ;i2
Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1

```

Page 1-2

```

0038  59                        pop cx ;i1
0039  83 F9 00                cmp cx, 0
003C  79 02                    jns ilabs ;|i2| = |i2|
003E  F7 D9                    neg cx
0040                                ilabs:
0040  B8 0001                  mov ax, k
0043  3D 0000                  cmp ax, 0
0046  75 0B                    jne f3v2
0048  83 F9 06                cmp cx, 6
004B  7C 0F                    jl finish
004D  B9 0006                  mov cx, 6
0050  EB 0A 90                  jmp finish

0053                                f3v2:

```

```

0053  83 FA 00                cmp dx, 0
0056  79 02                jns i2abs ; |i1| = |i1|
0058  F7 DA                neg dx
005A                      i2abs:
005A  03 CA                add cx, dx

005C                      finish:
005C  51                  push cx
005D  59                  pop cx
005E  CB                  ret
005F                      Main ENDP
005F                      CODE ENDS
                        END Main
Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1

```

Symbols-1

Segments and Groups:

N a m e	Length	Align
Combine Class		
ASTACK	0018	PARA
STACK		
CODE	005F	PARA
NONE		
DATA	0002	PARA
NONE		

Symbols:

N a m e	Type	Value
Attr		
A	NUMBER	-0003
B	NUMBER	0002
F3	L NEAR	0037
CODE		
F3V2	L NEAR	0053
CODE		

FINISH	L	NEAR	005C
CODE			
FLESS	L	NEAR	0022
CODE			
I	NUMBER		0002
I1ABS	L	NEAR	0040
CODE			
I2ABS	L	NEAR	005A
CODE			
K	NUMBER		0001
MAIN	F	PROC	0000
CODE Length = 005F			
RES	L	WORD	0000
DATA			
@CPU	TEXT	0101h	
@FILENAME	TEXT	LR3	
@VERSION	TEXT	510	

78 Source Lines
 78 Total Lines
 20 Symbols

47978 + 461329 Bytes symbol space free

0 Warning Errors
 0 Severe Errors