

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 = -7 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

```
a EQU -3
b EQU 2
i EQU 2
k EQU 0
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

DATA SEGMENT

    res DW 0
DATA ENDS

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

Main PROC FAR
    push ds
    sub ax, ax
    push ax

    mov ax, i
    shl ax, 1
    shl ax, 1
    mov bx, ax

    mov cx, a
    cmp cx, b
    jle fless

    neg ax
    add ax, 7
    push ax ;i1

    mov ax, bx
    neg ax
```

```

        add ax, 5
        push ax ;i2
        jmp f3
fless:
        mov ax, bx
        add ax, i
        add ax, i
        neg ax
        add ax, 8
        push ax

        mov ax, bx
        neg ax
        add ax, i
        add ax, 10
        push ax

f3:
        pop cx ;i1
        add cx, 0
        cmp cx, 0
        jns ilabs ; |i1| = |i1|
        neg cx
ilabs:
        mov ax, k
        cmp ax, 0
        jne f3v2
        pop bx
        cmp cx, 6
        jl finish
        mov cx, 6
        jmp finish

f3v2:
        pop dx ;i2
        add dx, 0
        cmp dx, 0
        jns i2abs ;|i2| = |i2|
        neg dx
i2abs:
        add cx, dx

finish:
        mov [res], cx

```

```

        push cx
        pop cx
        ret
Main ENDP
CODE ENDS
        END Main

```

ПРИЛОЖЕНИЕ В

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: LR3.lst

```

Microsoft      (R)      Macro      Assembler      Version      5.10
11/13/22 17:24:3

```

Page 1-1

```

=-0003          a EQU -3
= 0002          b EQU 2
= 0002          i EQU 2
= 0000          k EQU 0
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 11                  jle fless

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

001B  8B C3                  mov ax, bx
001D  F7 D8                  neg ax
001F  05 0005               add ax, 5
0022  50                    push ax ;i2
0023  EB 1A 90              jmp f3
0026                                fless:
0026  8B C3                  mov ax, bx
0028  05 0002               add ax, i
002B  05 0002               add ax, i
002E  F7 D8                  neg ax
0030  05 0008               add ax, 8
0033  50                    push ax

0034  8B C3                  mov ax, bx
0036  F7 D8                  neg ax
0038  05 0002               add ax, i
003B  05 000A               add ax, 10
Microsoft (R) Macro Assembler Version 5.10
11/13/22 17:24:3

Page 1-2

003E  50                    push ax

003F                                f3:
003F  59                    pop cx ;i1
0040  83 C1 00               add cx, 0
0043  83 F9 00               cmp cx, 0
0046  79 02                  jns ilabs ; |i1| = |i1|

```



```

0048  F7 D9                neg cx
004A                      ilabs:
004A  B8 0000                mov ax, k
004D  3D 0000                cmp ax, 0
0050  75 0C                jne f3v2
0052  5B                    pop bx
0053  83 F9 06                cmp cx, 6
0056  7C 13                jl finish
0058  B9 0006                mov cx, 6
005B  EB 0E 90                jmp finish

005E                      f3v2:
005E  5A                    pop dx ;i2
005F  83 C2 00                add dx, 0
0062  83 FA 00                cmp dx, 0
0065  79 02                jns i2abs ;|i2| = |i2|
0067  F7 DA                neg dx
0069                      i2abs:
0069  03 CA                add cx, dx

006B                      finish:
006B  89 0E 0000 R          mov [res], cx
006F  51                    push cx
0070  59                    pop cx
0071  CB                    ret
0072                      Main ENDP
0072                      CODE ENDS
                          END Main

Microsoft (R) Macro Assembler Version 5.10
11/13/22 17:24:3

```

Symbols-1

Segments and Groups:

	N a m e	Length	Align
Combine Class			
ASTACK	0018	PARA
STACK			
CODE	0072	PARA
NONE			

DATA 0002 PARA
NONE

Symbols:

Attr	N a m e	Type	Value
A		NUMBER	-0003
B		NUMBER	0002
F3		L NEAR	003F
CODE			
F3V2		L NEAR	005E
CODE			
FINISH		L NEAR	006B
CODE			
FLESS		L NEAR	0026
CODE			
I		NUMBER	0002
I1ABS		L NEAR	004A
CODE			
I2ABS		L NEAR	0069
CODE			
K		NUMBER	0000
MAIN		F PROC	0000
CODE	Length = 0072		
RES		L WORD	0000
DATA			
@CPU		TEXT	0101h
@FILENAME		TEXT	LR3
@VERSION		TEXT	510

86 Source Lines
86 Total Lines
20 Symbols

48018 + 461289 Bytes symbol space free

0 Warning Errors

0 Severe Errors