

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

Смирнов И.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Познакомиться с представлением и обработкой целых чисел на языке ассемблер.

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 .

Замечания:

1) при разработке программы нельзя использовать фрагменты, представленные на ЯВУ, в частности, для ввода-вывода данных. Исходные данные должны вводиться, а результаты контролироваться в режиме отладки;

2) при вычислении функций $f1$ и $f2$ вместо операции умножения следует использовать арифметический сдвиг и, возможно, сложение;

3) при вычислении функций $f1$ и $f2$ нельзя использовать процедуры;

4) при разработке программы следует минимизировать длину кода, для чего, если надо, следует преобразовать исходные выражения для вычисления функций.

Вариант 15 (шифр задания – 3.5.3):

$/7-4*i$, при $a>b$

$i1 = <$

$\backslash 8 - 6*i$, при $a \leq b$

$/20-4*i$, при $a>b$

$i2 = <$

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

$/ |i1+i2|$, при $k = 0$

$res = <$

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

Выполнение работы.

Для расчета значений функций использовались команды `cmp`, `jne`, `jle`, `jl`.

Для сложения использовалась команда `add`, а для умножения – битовый сдвиг влево `shl`.

Разработанный программный код см. в приложении А.

Текст файла диагностических сообщений см. в приложении Б.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

Значения	i1	i2	res	Комментарий
a = 2 b = 1 k = 0 i = 1	0003h = 3	0010h = 16	0013h = 19	ВЕРНО
a = 2	0003h = 3	0010h = 16	0003h = 3	ВЕРНО

b = 1 k = 1 i = 1				
a = 1 b = 2 k = 0 i = -2	0014h = 20	0012h = 18	0026h = 38	ВЕРНО
a = 1 b = 2 k = 1 i = -2	0014h = 20	0012h = 18	0012h = 18	ВЕРНО

Выводы.

В ходе выполнения данной лабораторной работы была изучена работа с целыми числами и условными переходами на языке Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr3.asm

```

AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
;Äàííûå ìðíãðàììû
DATA SEGMENT
;Äèðåèðèåû ììèñàíèý ààííûõ
a      DW      0
b      DW      0
i      DW      0
k      DW      0
i1     DW      0
i2     DW      0

DATA      ENDS

; Êîä ìðíãðàììû
CODE      SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack

; Ãìèíàíàý ìðíãðåóðå
Main      PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
    mov CX, 0

    mov a, 2
    mov b, 1
    mov i, 1
    mov k, 0

    ;â÷èñëåíèå f3
    mov cx, i
    shl cx, 1
    mov ax, cx
    shl cx, 1
    mov bx, b ;
    cmp a, bx ; ñåâåðåíèå a è b
    jle f1second
        neg cx
        add cx, 7
        jmp f1final
f1second:
    add cx, ax
    neg cx
    add cx, 8
f1final:
    mov i1, cx

    ;â÷èñëåíèå f5

```

```

mov cx, ax
shl cx, 1
cmp a, bx
jle f2second
    neg cx
    add cx, 20
    jmp f2final
f2second:
    add cx, ax
    add cx, -6
    neg cx
f2final:
mov i2, cx

; ðàññ÷âð res
mov bx, k
cmp bx, 0
jne resSecond
    add cx, i1
    cmp cx, 0
    jge MainFinal
    neg cx
    jmp MainFinal
resSecond:
    cmp cx, i1
    jl MainFinal
    mov cx, i1
MainFinal:
ret
; â cx ëâæèð çíà÷âíèâ res
Main      ENDP
CODE      ENDS
END Main

```

ПРИЛОЖЕНИЕ Б **ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ**

Название файла: lr3.lst

Microsoft (R) Macro Assembler Version 5.10
14:12:46

12/9/21

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

; ?????????????????
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 ---- R     mov   AX,DATA
0007  8E D8          mov   DS,AX
0009  B9 0000        mov   CX, 0

000C  C7 06 0000 R 0002      mov a, 2
0012  C7 06 0002 R 0001      mov b, 1
0018  C7 06 0004 R 0001      mov i, 1
001E  C7 06 0006 R 0000      mov k, 0

; ????????????????? f3
0024  8B 0E 0004 R          mov cx, i
0028  D1 E1                shl  cx, 1
002A  8B C1                mov  ax, cx
002C  D1 E1                shl  cx, 1
002E  8B 1E 0002 R          mov  bx, b      ;
0032  39 1E 0000 R          cmp  a, bx      ; ????????????? a
; b
0036  7E 08                jle  f1second

```

```

0038 F7 D9                neg cx
003A 83 C1 07            add cx, 7
003D EB 08 90            jmp f1final
0040                f1second:
0040 03 C8                add cx, ax
0042 F7 D9                neg cx
0044 83 C1 08            add cx, 8
0047                f1final:
0047 89 0E 0008 R        mov i1, cx

```

Microsoft (R) Macro Assembler Version 5.10
14:12:46

12/9/21

Page

1-2

```

                                ;???????????? f5
004B 8B C8                mov cx, ax
004D D1 E1                shl cx, 1
004F 39 1E 0000 R        cmp a, bx
0053 7E 08                jle f2second
0055 F7 D9                neg cx
0057 83 C1 14            add cx, 20
005A EB 08 90            jmp f2final
005D                f2second:
005D 03 C8                add cx, ax
005F 83 C1 FA            add cx, -6
0062 F7 D9                neg cx
0064                f2final:
0064 89 0E 000A R        mov i2, cx

                                ;???????? res
0068 8B 1E 0006 R        mov bx, k
006C 83 FB 00            cmp bx, 0
006F 75 0E                jne resSecond
0071 03 0E 0008 R        add cx, i1
0075 83 F9 00            cmp cx, 0
0078 7D 0F                jge MainFinal
007A F7 D9                neg cx
007C EB 0B 90            jmp MainFinal
007F                resSecond:
007F 3B 0E 0008 R        cmp cx, i1
0083 7C 04                jl MainFinal
0085 8B 0E 0008 R        mov cx, i1
0089                MainFinal:                                ; ? cx ??????

?????                                ?????? res
0089 CB                ret
008A                Main                ENDP
008A                CODE                ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10
14:12:46

12/9/21

s-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK		0018	PARA	STACK
CODE		008A	PARA	NONE
DATA		000C	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
A		L WORD	0000	DATA
B		L WORD	0002	DATA
F1FINAL		L NEAR	0047	CODE
F1SECOND		L NEAR	0040	CODE
F2FINAL		L NEAR	0064	CODE
F2SECOND		L NEAR	005D	CODE
I		L WORD	0004	DATA
I1		L WORD	0008	DATA
I2		L WORD	000A	DATA
K		L WORD	0006	DATA
008A MAIN		F PROC	0000	CODE Length =
MAINFINAL		L NEAR	0089	CODE
RESSECOND		L NEAR	007F	CODE
@CPU		TEXT	0101h	
@FILENAME		TEXT	1r3	
@VERSION		TEXT	510	

84 Source Lines
84 Total Lines
21 Symbols

47996 + 461311 Bytes symbol space free

0 Warning Errors
0 Severe Errors