

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

Бояркин Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить механизм работы представлений и обработки целых чисел.

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 22

$$f4 = \begin{cases} / -(6*i - 4), & \text{при } a > b \\ \backslash 3*(i+2), & \text{при } a \leq b \end{cases}$$

$$f8 = \begin{cases} / -(6*i+8), & \text{при } a > b \\ \backslash 9 - 3*(i-1), & \text{при } a \leq b \end{cases}$$

$$f3 = \begin{cases} / |i1 + i2|, & \text{при } k=0 \\ \backslash \min(i1,i2), & \text{при } k \neq 0 \end{cases}$$

Выполнение работы.

Происходит расчет функций f_1 , f_2 , f_3 . При организации ветвящихся процессов использовалась функция `str` и условные переходы. Для операций умножения использовался побитовый сдвиг влево и сложение.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Проверка работы программы с помощью отладчика (все результаты заносились в реестр CX)

№ п/п	Входные данные	Полученное значение i_1	Полученное значение i_2	Полученное значение res	Комментарии
1	$a = 1$ $b = 2$ $i = 3$ $k = 4$	$000Fh = 15$	$0003h = 3$	$0003h = 3$	Верно, т.к. $a \leq b$, $3*(3+2) = 15$, $9 - 3*(3-1) = 3$, $k \neq 0$, $\min(15,3) = 3$
2	$a = 1$ $b = 2$ $i = 3$ $k = 0$	$000Fh = 15$	$0003h = 3$	$0012h = 18$	Верно, т.к. $a \leq b$, $3*(3+2) = 15$, $9 - 3*(3-1) = 3$, $k = 0$, $ 15+3 = 18$
3	$a = 2$ $b = 1$ $i = 3$ $k = 0$	$FFF2h = -14$	$FFE6h = -26$	$0028h = 40$	Верно, т.к. $a > b$, $-(6*3 - 4) = -14$, $-(6*3 + 8) = -26$, $k = 0$, $ -14 + (-26) = 40$
4	$a = 2$ $b = 1$ $i = 3$ $k = -1$	$FFF2h = -14$	$FFE6h = -26$	$FFE6h = -26$	Верно, т.к. $a > b$, $-(6*3 - 4) = -14$, $-(6*3 + 8) = -26$, $k \neq 0$, $\min(-14, -26) = -26$

Выводы.

Был изучен механизм работы представлений и обработки целых чисел.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.asm

```
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

; Данные программы
DATA SEGMENT
; Директивы описания данных
string DB 15, 15 DUP('$')
sign DB 1
a DW 1
b DW 2
i DW 3
k DW 4
i1 DW 0
i2 DW 0

DATA ENDS

; Код программы
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
Main PROC FAR
    push DS
    sub AX, AX
    push AX
    mov AX, DATA
    mov DS, AX
    mov CX, 0

; вычисление f4
    mov cx, i
    mov ax, cx
    mov bx, b
    cmp a, bx
    jle f4second ; a != bx
    shl cx, 1
    mov ax, cx
    shl cx, 1
    add cx, ax
    add cx, -4
    neg cx
    jmp f4final
f4second: ; a = bx
    add cx, 2
```

```
    mov ax, cx
    shl cx, 1
    add cx, ax
f4final:
mov i1, cx
```

```
    ;вычисление f8
    mov cx, i
    mov ax, cx
    cmp a, bx
    jle f8second ; a != bx
        shl cx, 1
    mov ax, cx
    shl cx, 1
    add cx, ax
    add cx, 8
        neg cx
        jmp f8final
f8second: ; a = bx
        add cx, -1
        mov ax, cx
    shl cx, 1
    add cx, ax
        neg cx
        add cx, 9
f8final:
    mov i2, cx
```

```
;рассчет f3
mov bx, k
cmp bx, 0
je f3Second ; k != 0
    mov bx, i1
    cmp bx, i2
    jle min1
        mov cx, i2 ; i2 <= i1
        jmp MainFinal
min1:
    mov cx, bx ; i2 > i1
    jmp MainFinal
```

```
f3Second: ; k = 0
    mov cx, i1
    add cx, i2
    cmp cx, 0
    jge skip1 ;модуль i1 + i2
    neg cx
    mov i1, cx
    jmp MainFinal
```

```

        skip1:
            jmp MainFinal

MainFinal:
            ; в сx лежит значение функции f3
            ret
Main      ENDP
CODE     ENDS
END Main

```

Название файла: lab3.lst

```

Microsoft      (R)      Macro      Assembler      Version      5.10
11/15/21 23:37:4

```

Page

1-1

```

                                ; Стек программы
0000                                AStack SEGMENT  STACK
0000  000C[                                DW 12 DUP(?)
    ????                                ]

0018                                AStack  ENDS
                                ;Данные программы
0000                                DATA      SEGMENT
                                ;Директивы описания данны
                                x
0000  0F                                string DB    15, 15 DUP('$')
    000F[
    24                                ]

0010  01                                sign    DB    1
0011  0001                                a        DW    1
0013  0002                                b        DW    2
0015  0003                                i        DW    3
0017  0004                                k        DW    4
0019  0000                                i1       DW    0
001B  0000                                i2       DW    0

001D                                DATA      ENDS

                                ; Код программы
0000                                CODE      SEGMENT
                                ASSUME CS:CODE, DS:DATA, SS:AStack

                                ; Головная процедура
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

```

;вычисление f4

```

000C 8B 0E 0015 R   mov  cx, i
0010 8B C1          mov  ax, cx
0012 8B 1E 0013 R   mov  bx, b
0016 39 1E 0011 R   cmp  a, bx
001A 7E 10          jle  f4second ; a != bx
001C D1 E1          shl  cx, 1
001E 8B C1          mov  ax, cx
0020 D1 E1          shl  cx, 1
0022 03 C8          add  cx, ax
0024 83 C1 FC       add  cx, -4
0027 F7 D9          neg  cx
0029 EB 0A 90       jmp  f4final

```

```

Microsoft      (R)      Macro      Assembler      Version      5.10
11/15/21 23:37:4

```

Page

1-2

```

002C          f4second:      ; a = bx
002C 83 C1 02       add  cx, 2
002F 8B C1          mov  ax, cx
0031 D1 E1          shl  cx, 1
0033 03 C8          add  cx, ax
0035          f4final:
0035 89 0E 0019 R   mov  i1, cx

```

;вычисление f8

```

0039 8B 0E 0015 R   mov  cx, i
003D 8B C1          mov  ax, cx
003F 39 1E 0011 R   cmp  a, bx
0043 7E 10          jle  f8second ; a != bx
0045 D1 E1          shl  cx, 1
0047 8B C1          mov  ax, cx
0049 D1 E1          shl  cx, 1
004B 03 C8          add  cx, ax
004D 83 C1 08       add  cx, 8
0050 F7 D9          neg  cx
0052 EB 0F 90       jmp  f8final
0055          f8second:      ; a = bx
0055 83 C1 FF       add  cx, -1
0058 8B C1          mov  ax, cx
005A D1 E1          shl  cx, 1
005C 03 C8          add  cx, ax
005E F7 D9          neg  cx
0060 83 C1 09       add  cx, 9
0063          f8final:
0063 89 0E 001B R   mov  i2, cx

```



```

;расчет f3
0067 8B 1E 0017 R      mov bx, k
006B 83 FB 00          cmp bx, 0
006E 74 16            je f3Second ; k != 0
0070 8B 1E 0019 R      mov bx, i1
0074 3B 1E 001B R      cmp bx, i2
0078 7E 07            jle min1
007A 8B 0E 001B R      mov cx, i2      ; i2 <= i1
007E EB 1F 90          jmp MainFinal
0081                  min1:
0081 8B CB            mov cx, bx      ; i2 > i1
0083 EB 1A 90          jmp MainFinal

```

```

0086                  f3Second: ; k = 0
0086 8B 0E 0019 R      mov cx, i1
008A 03 0E 001B R      add cx, i2
008E 83 F9 00          cmp cx, 0
0091 7D 09            jge skip1      ;модуль i1 + i2
0093 F7 D9            neg cx
0095 89 0E 0019 R      mov i1, cx

```

Microsoft (R) Macro Assembler Version 5.10
11/15/21 23:37:4

Page

1-3

```

0099 EB 04 90          jmp MainFinal
009C                  skip1:
009C EB 01 90          jmp MainFinal

```

```

009F                  MainFinal:      ; в cx лежи
                                т значение функции f3
009F CB              ret
00A0                  Main          ENDP
00A0                  CODE          ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10
11/15/21 23:37:4

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
	ASTACK	0018	PARA	STACK
	CODE	00A0	PARA	NONE

DATA 001D PARA NONE

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0011	DATA
B	L WORD	0013	DATA
F3SECOND	L NEAR	0086	CODE
F4FINAL	L NEAR	0035	CODE
F4SECOND	L NEAR	002C	CODE
F8FINAL	L NEAR	0063	CODE
F8SECOND	L NEAR	0055	CODE
I	L WORD	0015	DATA
I1	L WORD	0019	DATA
I2	L WORD	001B	DATA
K	L WORD	0017	DATA
MAIN	F PROC	0000	CODE Length
MAINFINAL	L NEAR	009F	CODE
MIN1	L NEAR	0081	CODE
SIGN	L BYTE	0010	DATA
SKIP1	L NEAR	009C	CODE
STRING	L BYTE	0000	DATA
@CPU	TEXT	0101h	
@FILENAME	TEXT	lab3	
@VERSION	TEXT	510	

= 00A0

109 Source Lines
109 Total Lines
25 Symbols

47962 + 459298 Bytes symbol space free

0 Warning Errors
0 Severe Errors