

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
a          DW      2
b          DW      1
i          DW      3
k          DW      0
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
    shl    ax, 1
    add    ax, cx ; в ax = 3i
    mov    bx, b
    cmp    a, bx

    jle    ifless
    shl    ax, 1
    mov    cx, ax
    sub    ax, 4
    neg    ax
    mov    i1, ax
    add    cx, 8
    neg    cx
    mov    i2, cx
```

```

        jmp finfun

ifless:
    mov cx, ax
    add ax, 6
    mov i1, ax
    mov ax, cx
    mov cx, 12
    sub cx, ax
    mov i2, cx

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

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

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

```

Название файла: lab3.lst

Microsoft	(R)	Macro	Assembler	Version	5.10
11/23/21 15:58:5					Page
1-1					

```

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

0018                                AStack  ENDS
                                ;Данные программы
0000                                DATA      SEGMENT

```

```

;Директивы описания даннь
x
0000 0002          a      DW      2
0002 0001          b      DW      1
0004 0003          i      DW      3
0006 0000          k      DW      0
0008 0000          i1     DW      0
000A 0000          i2     DW      0

000C                      DATA      ENDS

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

; Главная 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 0004 R                  mov cx, i
0010 8B C1                      mov ax, cx
0012 D1 E0                      shl ax, 1
0014 03 C1                      add ax, cx ; в ax = 3i
0016 8B 1E 0002 R                  mov bx, b
001A 39 1E 0000 R                  cmp a, bx

001E 7E 18                      jle ifless
0020 D1 E0                      shl ax, 1
0022 8B C8                      mov cx, ax
0024 2D 0004                      sub ax, 4
0027 F7 D8                      neg ax
0029 A3 0008 R                  mov i1, ax
002C 83 C1 08                      add cx, 8
002F F7 D9                      neg cx
0031 89 0E 000A R                  mov i2, cx
0035 EB 14 90                      jmp finfun

```

```

Microsoft      (R)      Macro      Assembler      Version      5.10
11/23/21 15:58:5

```

1-2

Page

```

0038                      ifless:
0038 8B C8                      mov cx, ax

```

```

003A 05 0006                add ax, 6
003D A3 0008 R              mov i1, ax
0040 8B C1                  mov ax, cx
0042 B9 000C                  mov cx, 12
0045 2B C8                  sub cx, ax
0047 89 0E 000A R            mov i2, cx

                                ;расчет f3
004B                        finfun:
004B 8B 1E 0006 R            mov bx, k
004F 83 FB 00                cmp bx, 0
0052 74 10                  je f3Second ; k != 0
0054 3B 0E 0008 R            cmp cx, i1
0058 7E 07                  jle min1
005A 8B 0E 0008 R            mov cx, i1          ; i2 <= i1
005E EB 12 90                jmp MainFinal
0061                        min1:
0061 EB 0F 90                jmp MainFinal

0064                        f3Second: ; k = 0
0064 03 0E 0008 R            add cx, i1
0068 83 F9 00                cmp cx, 0
006B 7D 05                  jge MainFinal      ;модуль i1
                                + i2
006D F7 D9                  neg cx
006F EB 01 90                jmp MainFinal

0072                        MainFinal:          ; в cx лежи
                                т значение функции f3
0072 CB                      ret
0073                        Main                ENDP
0073                        CODE                ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10
11/23/21 15:58:5

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine
Class				
	ASTACK	0018	PARA	STACK
	CODE	0073	PARA	NONE
	DATA	000C	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
A	L WORD	0000	DATA

B	L WORD	0002	DATA
F3SECOND	L NEAR	0064	CODE
FINFUN	L NEAR	004B	CODE
I	L WORD	0004	DATA
I1	L WORD	0008	DATA
I2	L WORD	000A	DATA
IFLESS	L NEAR	0038	CODE
K	L WORD	0006	DATA
MAIN	F PROC	0000	CODE Length
= 0073			
MAINFINAL	L NEAR	0072	CODE
MIN1	L NEAR	0061	CODE
@CPU	TEXT	0101h	
@FILENAME	TEXT	lab3	
@VERSION	TEXT	510	

83 Source Lines
83 Total Lines
20 Symbols

48040 + 461267 Bytes symbol space free

0 Warning Errors
0 Severe Errors