

**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

Самара Р.Д.

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 13

$$f2 = \begin{cases} / - (4*i+3) , & \text{при } a > b \\ \backslash 6*i - 10 , & \text{при } a \leq b \end{cases}$$

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

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

### 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)

| №<br>п/п | Входные данные                              | Полученное<br>значение $i_1$ | Полученное<br>значение $i_2$ | Полученное<br>значение $res$ | Комментарии   |
|----------|---|------------------------------|------------------------------|------------------------------|---|
| 1        | $a = 1$<br>$b = 2$<br>$i = 3$<br>$k = 4$    | $0008h = 8$                  | $0003h = 3$                  | $0003h = 3$                  | Верно ( $a < b \Rightarrow$<br>$f_2 = 6 * 3 - 10 = 8$ ;<br>$a < b \Rightarrow f_8 = 9 - 6 = 3$ ;<br>$k \neq 0 \Rightarrow$<br>$\min(8, 3) = 3$ ). |
| 2        | $a = 1$<br>$b = 2$<br>$i = 3$<br>$k = 0$    | $0008h = 8$                  | $0003h = 3$                  | $000Bh = 11$                 | Верно   |
| 3        | $a = -3$<br>$b = -2$<br>$i = -1$<br>$k = 1$ | $FFF0h = -16$                | $000Fh = 15$                 | $FFF0h = -16$                | Верно   |
| 4        | $a = 2$<br>$b = 1$<br>$i = 3$<br>$k = 0$    | $FFF1h = -15$                | $FFE6h = -26$                | $0029h = 41$                 | Верно   |

### Вывод.

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      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 f2
    mov  CX, i
    mov  AX, CX
    shl  CX, 1
    shl  CX, 1
    mov  BX, b
    cmp  a, bx    ; сравнение a и b
    jle  f2second
    add  CX, 3    ; a > b
    neg  CX
    jmp  f2final
f2second:                ; a <= b
    add  CX, AX
    add  CX, AX
    add  CX, -10
f2final:
    mov  i1, CX

    ; вычисление f8
    mov  CX, i
    mov  AX, CX
    cmp  a, bx
    jle  f8second
```

```

        shl cx, 1      ; a > bx
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

;вычисление 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      ; |i1 + i2|
        neg cx
        mov i1, cx
        jmp MainFinal
skip1:
        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  
24:37:2

11/17/21

Page

1-1

```

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

```

]

```
0018          AStack ENDS
;Данные программы
0000          DATA      SEGMENT
;Директивы описания данных
0000 0001          a      DW      1
0002 0002          b      DW      2
0004 0003          i      DW      3
0006 0004          k      DW      4
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 f2
000C 8B 0E 0004 R      mov cx, i
0010 8B C1          mov ax, cx
0012 D1 E1          shl cx, 1
0014 D1 E1          shl cx, 1
0016 8B 1E 0002 R      mov bx, b
001A 39 1E 0000 R      cmp a, bx      ; сравнение a и b
001E 7E 08          jle f2second
0020 83 C1 03          add cx, 3      ;a > b
0023 F7 D9          neg cx
0025 EB 08 90          jmp f2final
0028          f2second:          ;a <= b
0028 03 C8          add cx, ax
002A 03 C8          add cx, ax
002C 83 C1 F6          add cx, -10
002F          f2final:
002F 89 0E 0008 R      mov i1, cx

;вычисление f8
0033 8B 0E 0004 R      mov cx, i
0037 8B C1          mov ax, cx
0039 39 1E 0000 R      cmp a, bx

Microsoft (R) Macro Assembler Version 5.10
24:37:2
1-2
003D 7E 10          jle f8second
003F D1 E1          shl cx, 1      ; a > bx
```

11/17/21

Page

```

0041 8B C1          mov ax, cx
0043 D1 E1          shl cx, 1
0045 03 C8          add cx, ax
0047 83 C1 08       add cx, 8
004A F7 D9          neg cx
004C EB 0F 90       jmp f8final
004F              f8second: ; a <= bx
004F 83 C1 FF       add cx, -1
0052 8B C1          mov ax, cx
0054 D1 E1          shl cx, 1
0056 03 C8          add cx, ax
0058 F7 D9          neg cx
005A 83 C1 09       add cx, 9
005D              f8final:
005D 89 0E 000A R   mov i2, cx

;вычисление f3
0061 8B 1E 0006 R   mov bx, k
0065 83 FB 00       cmp bx, 0
0068 74 16          je f3Second ; k != 0
006A 8B 1E 0008 R   mov bx, i1
006E 3B 1E 000A R   cmp bx, i2
0072 7E 07          jle min1
0074 8B 0E 000A R   mov cx, i2 ; i2 <= i1
0078 EB 1F 90       jmp MainFinal
007B              min1:
007B 8B CB          mov cx, bx ; i2 > i1
007D EB 1A 90       jmp MainFinal
0080              f3Second: ; k = 0
0080 8B 0E 0008 R   mov cx, i1
0084 03 0E 000A R   add cx, i2
0088 83 F9 00       cmp cx, 0
008B 7D 09          jge skip1 ; |i1 + i2|
008D F7 D9          neg cx
008F 89 0E 0008 R   mov i1, cx
0093 EB 04 90       jmp MainFinal
0096              skip1:
0096 EB 01 90       jmp MainFinal

0099              MainFinal: ; в cx лежит зна
                     чение функции f3
0099 CB           ret
009A              Main      ENDP
009A              CODE      ENDS
                     END Main

```

Microsoft (R) Macro Assembler Version 5.10  
24:37:2

11/17/21

Symbols-1

#### Segments and Groups:

|        | N a m e   | Length | Align | Combine Class |
|--------|-----------|--------|-------|---------------|
| ASTACK | . . . . . | 0018   | PARA  | STACK         |
| CODE   | . . . . . | 009A   | PARA  | NONE          |
| DATA   | . . . . . | 000C   | PARA  | NONE          |

Symbols:

|        | N a m e             | Type   | Value | Attr |        |
|--------|---------------------|--------|-------|------|--------|
|        | A . . . . .         | L WORD | 0000  | DATA |        |
|        | B . . . . .         | L WORD | 0002  | DATA |        |
|        | F2FINAL . . . . .   | L NEAR | 002F  | CODE |        |
|        | F2SECOND . . . . .  | L NEAR | 0028  | CODE |        |
|        | F3SECOND . . . . .  | L NEAR | 0080  | CODE |        |
|        | F8FINAL . . . . .   | L NEAR | 005D  | CODE |        |
|        | F8SECOND . . . . .  | L NEAR | 004F  | CODE |        |
|        | I . . . . .         | L WORD | 0004  | DATA |        |
|        | I1 . . . . .        | L WORD | 0008  | DATA |        |
|        | I2 . . . . .        | L WORD | 000A  | DATA |        |
|        | K . . . . .         | L WORD | 0006  | DATA |        |
| = 009A | MAIN . . . . .      | F PROC | 0000  | CODE | Length |
|        | MAINFINAL . . . . . | L NEAR | 0099  | CODE |        |
|        | MIN1 . . . . .      | L NEAR | 007B  | CODE |        |
|        | SKIP1 . . . . .     | L NEAR | 0096  | CODE |        |
|        | @CPU . . . . .      | TEXT   | 0101h |      |        |
|        | @FILENAME . . . . . | TEXT   | 1b3   |      |        |
|        | @VERSION . . . . .  | TEXT   | 510   |      |        |

97 Source Lines  
97 Total Lines  
23 Symbols

47992 + 461315 Bytes symbol space free

0 Warning Errors  
0 Severe Errors