



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

**КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»**

ЛАБОРАТОРНАЯ РАБОТА №3

«Выполнение арифметических операций над числами без знака и со знаком»

ДИСЦИПЛИНА: «Системное программирование»

Выполнил: студент гр. ИУК4-31Б


(подпись)

(Суриков Н. С.)
(Ф.И.О.)

Проверил:


(подпись)

(Амеличева К. А.)
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Цель: научиться выполнять операции сложения, вычитания, умножения и деления с целыми числами без знака и со знаком.

Вариант 8

Листинг файла макросов:

```
1  ; Макрос для вычисления  $y = 2*b + 2*a$ 
2  mCalc_Y1 MACRO a, b
3      push bx          ; Данные в стек
4      push cx
5      push dx
6
7      mov ax, a        ; загрузить a в ax
8      mov bx, b        ; загрузить b в bx
9      shl ax, 1        ;  $ax = 2*a$ 
10     shl bx, 1        ;  $bx = 2*b$ 
11     add ax, bx        ;  $ax = 2*a + 2*b$ 
12
13     pop dx
14     pop cx
15     pop bx
16 ENDM
17
18 ; Макрос для вычисления  $y = ((a + 3*b) / c) + 4$ 
19 mCalc_Y2 MACRO a, b, c
20     push bx          ; Данные в стек
21     push cx
22     push dx
23
24     mov bx, a        ; загрузить a в bx
25     mov ax, b        ; загрузить b в ax
26     mov cx, 3        ; загрузить 3 в cx
27
28     imul cx          ;  $ax = b * 3$  ( $ax = 3*b$ )
29     add ax, bx        ;  $ax = a + 3*b$ 
30
31     xor dx, dx        ; очистить для деления
32     xor cx, cx
33
34     mov cx, c        ; загрузить c в cx
35     cmp cx, 0
36
37     je DIV_BY_ZERO   ; если  $c = 0$ , переход к обработке деления на ноль
38     cwd
39     idiv cx          ; деление ax на c, результат в ax/dx
40     add ax, 4        ; добавить 4 к результату
41     jmp END_CALC_Y2
42 DIV_BY_ZERO:
43     xor ax, ax        ; если деление на ноль, сохранить 0 в res
44 END_CALC_Y2:
45     pop dx
46     pop cx
47     pop bx
48 ENDM
49
50 ; Макрос ввода 10-чного числа в регистр AX
51 mReadAX macro buffer, size
52     local input, startOfConvert, endOfConvert
53     push bx          ; Данные в стек
54     push cx
55     push dx
56 input:
57     mov [buffer], size ; Задаём размер буфера
```

```

58      mov     dx, offset [buffer]                ; Поместить в регистр dx строку по
адресу buffer
59      mov     ah, 0Ah                          ; Чтение строки из консоли
60      int     21h                              ; Прерывание DOS
61
62      mov     ah, 02h                          ; Вывод символа на экран
63      mov     dl, 0Bh                          ; Перевод каретки на новую строку
64      int     21h                              ; Прерывание DOS
65
66      mov     ah, 02h                          ; Вывод символа на экран
67      mov     dl, 0Ah                          ; Чтение строки из консоли
68      int     21h                              ; Прерывание DOS
69
70      xor     ah, ah                            ; Очистка регистра ah
71      cmp     al, [buffer][1]                  ; Проверка на пустую строку
72      jz      input                            ; Переход, если строка пустая
73
74      xor     cx, cx                            ; Очистка регистра cx
75      mov     cl, [buffer][1]                  ; инициализация переменной-
счётчика
76
77      xor     ax, ax                            ; Очистка регистра ax
78      xor     bx, bx                            ; Очистка регистра bx
79      xor     dx, dx                            ; Очистка регистра dx
80
81      mov     bx, offset [buffer][2]            ; Поместить начало строки в
регистр bx
82      cmp     [buffer][2], '-'                 ; Проверка на знак числа
83      jne     startOfConvert                    ; Переход, если число
неотрицательное
84      inc     bx                                ; Инкремент регистра bx
85      dec     cl                                ; Декремент регистра-счётчика cl
86      startOfConvert:
87      mov     dx, 10                          ; Поместить в регистр ax число 10
88      mul     dx                              ; Умножение на 10 перед сложением
с младшим разрядом
89      cmp     ax, 8000h                       ; Проверка числа на выход за
границы
90      jae     input                            ; Переход, если число выходит за
границы
91      mov     dl, [bx]                        ; Поместить в регистр dl следующий
символ
92      sub     dl, '0'                         ; Перевод его в числовой формат
93      add     ax, dx                          ; Прибавляем его к конечному
результату
94      cmp     ax, 8000h                       ; Проверка числа на выход за
границы
95      jae     input                            ; Переход, если число выходит за
границы
96      inc     bx                                ; Переход к следующему символу
97      loop    startOfConvert                  ; Цикл
98      cmp     [buffer][2], '-'                 ; Проверка на знак числа
99      jne     endOfConvert                    ; Переход, если число
неотрицательное
100     neg     ax                              ; Инвертирование числа
101     endOfConvert:
102     pop     dx                                ; Данные из стека
103     pop     cx
104     pop     bx
105     endm
106
107 ; Макрос вывода 10-чного числа из регистра AX
108 mWriteAX macro
109     local convert, write
110     push    ax                                ; Данные в стек
111     push    bx
112     push    cx
113     push    dx
114     push    di
115

```

```

116         mov     cx, 10             ; cx - основание системы счисления
117         xor     di, di             ; di - количество цифр в числе
118         or      ax, ax             ; Проверка числа на ноль
119         jns     convert            ; Переход, если число положительное
120         push    ax                 ; Регистр ax в стек
121         mov     dx, '-'            ; Поместить в регистр dx символ '-'
122         mov     ah, 02h            ; Вывод символа на экран
123         int     21h                ; Прерывание DOS
124         pop     ax                 ; Регистр ax из стека
125         neg     ax                 ; Инвертирование отрицательного числа
126     convert:
127         xor     dx, dx             ; Очистка регистра dx
128         div     cx                 ; После деления dl = остатку от деления ax на cx
129         add     dl, '0'            ; Перевод в символьный формат
130         inc     di                 ; Увеличение количества цифр в числе на 1
131         push    dx                 ; Регистр dx в стек
132         or      ax, ax             ; Проверка числа на ноль
133         jnz     convert            ; Переход, если число не равно нулю
134     write:
135         pop     dx                 ; dl = очередной символ
136         mov     ah, 02h            ; Вывод символа на экран
137         int     21h                ; Прерывание DOS
138         dec     di                 ; Повторение, пока di != 0
139         jnz     write
140
141         pop     di                 ; Данные из стека
142         pop     dx
143         pop     cx
144         pop     bx
145         pop     ax
146     endm
147
148 ; Макрос вывода строки
149 mWriteStr macro string
150         push    ax
151         push    dx
152
153         mov     ah, 09h
154         mov     dx, offset string
155         int     21h
156
157         pop     dx
158         pop     ax
159     endm

```

Листинг программы Lr3_Prog1:

```

1  ; var8: x1=92 x2=a1h=161 x3=222 a=39 b=33h=51 c=5 d=6
2  include macroses.asm
3  .model small
4  .stack 100h
5  .data
6      x1      db 92
7      x2      db 161
8      x3      db 222
9      a       db 39
10     b       db 51
11     c       db 5
12     d       db 6
13     e       db ?
14     f       db ?
15     sum_result db 0

```


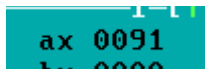

```

16     sub_result db 0
17     mul_result dw 1
18     rem_result db 0
19     div_result db 0
20     mes_res db 'Result: ', '$'
21     new_line db 13,10,'$'
22
23 .code
24 calc_value macro reg
25         mov al, reg
26         sub al, a
27         mov sub_result, al
28         add al, b
29         mov sum_result, al
30         mul c
31         mov mul_result, ax
32         div d
33         inc al
34         mov e, al
35         dec ah
36         mov f, ah
37         add al, ah
38         xor ah, ah
39 endm
40 start:
41     mov     ax, @data
42     mov     ds, ax
43
44     xor     ax, ax
45     calc_value x1
46     mWriteStr mes_res
47     mWriteAX
48     mWriteStr new_line
49     xor     ax, ax
50     calc_value x2
51     mWriteStr mes_res
52     mWriteAX
53     mWriteStr new_line
54     xor     ax, ax
55     calc_value x3
56     mWriteStr mes_res
57     mWriteAX
58     mWriteStr new_line
59
60     mov     ax, 4C00h
61     int     21h
62 end start
63 end

```

Результат выполнения программы:

```
D:\>lr_prg1
Result: 90
Result: 145
Result: 195
```

X1: , X2: , X3: 

Листинг программы Lr3_Prog2:

```
1  include macroses.asm
2  .model small
3  .stack 100h
4  .data
5      buffer    db 5, 0, 5 dup(0)           ; Буфер для ввода числа (максимум 5 цифр)
6      mes_a     db 'Enter the number a: ', '$'
7      mes_b     db 'Enter the number b: ', '$'
8      mes_c     db 'Enter the number c: ', '$'
9      mes_d     db 'Enter the number d: ', '$'
10     mes_res1   db 'Result y1: ', '$'
11     mes_res2   db 'Result y2: ', '$'
12     new_line   db 13,10,'$'
13     a         dw 0                       ; Переменная для хранения первого числа
14     b         dw 0                       ; Переменная для хранения второго числа
15     c         dw 0                       ; Переменная для хранения третьего числа
16     d         dw 0                       ; Переменная для хранения четвертого числа
17
18 .code
19     start:
20         mov     ax, @data
21         mov     ds, ax
22
23     ; Ввод переменной a
24         xor     ax, ax
25         mWriteStr mes_a
26         mReadAX buffer, 5
27         mov     a, ax
28
29     ; Ввод переменной b
30         xor     ax, ax
31         mWriteStr mes_b
32         mReadAX buffer, 5
33         mov     b, ax
34
35     ; Ввод переменной c
36         xor     ax, ax
37         mWriteStr mes_c
38         mReadAX buffer, 5
39         mov     c, ax
40
41     ; Ввод переменной d
42         xor     ax, ax
43         mWriteStr mes_d
44         mReadAX buffer, 5
45         mov     d, ax
46
47
48         xor     ax, ax
49
50
51     mCalc_Y1 a, b           ; Пример 1: y = 2*b + 2*a
52     mWriteStr mes_res1
53     mWriteAX
54     mWriteStr new_line     ; Вывод результата
```

```

55
56         xor      ax, ax
57
58         mCalc_Y2 a, b, c      ; Пример 2:  $y = ((a + 3*b) / c) + 4$ 
59         mWriteStr mes_res2
60         mWriteAX
61         mWriteStr new_line    ; Вывод результата
62
63     ; Завершение программы
64         mov      ax, 4c00h
65         int      21h
66 end start

```

Результат выполнения программы:

<pre> D:\>lr_prg2 Enter the number a: 5 Enter the number b: 10 Enter the number c: 15 Enter the number d: 8 Result y1: 30 Result y2: 6 </pre>	<pre> D:\>lr_prg2 Enter the number a: -5 Enter the number b: -10 Enter the number c: -15 Enter the number d: -3 Result y1: -30 Result y2: 6 </pre>	<pre> D:\>lr_prg2 Enter the number a: -10 Enter the number b: -20 Enter the number c: 15 Enter the number d: 18 Result y1: -60 Result y2: 0 </pre>
<pre> D:\>lr_prg2 Enter the number a: 10 Enter the number b: -20 Enter the number c: -15 Enter the number d: 18 Result y1: -20 Result y2: 7 </pre>		

Вариант 3		1	2	3	4
Значения	a	5	-5	-10	10
	b	10	-10	-20	-20
	c	15	-15	15	-15
	d	8	-3	18	18
AX		0002	0002	FFFC	0003
DX		0005	FFFB	FFF6	FFFB

Вывод: в ходе работы были практически освоены навыки выполнения операций сложения, вычитания, умножения и деления с целыми числами без знака и со знаком