

**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**

Студентка гр. 0383

Александрович В.П.

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 1 (шифр задания – 1.2.1):

$/15-2*i$ , при  $a>b$

$i1 = <$

$\backslash 3*i+4$ , при  $a\leq b$

$/-(4*i+3)$ , при  $a>b$

$i2 = <$

$\backslash 6*i-10$ , при  $a\leq b$

$/\min(i1, i2)$ , при  $k = 0$

$res = <$

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

### Выполнение работы.

Выделяется память под хранение начальных данных. Для расчета функций используются команды `str`, `jne`, `jle`, `jl`. Для сложения используется команда `add`, а для умножения – битовый сдвиг влево.

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	0011h = 13	0001h = 1	0001h = 1	ВЕРНО
a = 2	0011h = 13	0001h = 1	0011h = 17	ВЕРНО

b = 1 k = 1 i = -1				
a = 1 b = 2 k = 0 i = 2	000Ah = 10	0002h = 2	0002h = 2	ВЕРНО
a = 1 b = 2 k = 1 i = 2	000Ah = 10	0002h = 2	000Ah = 10	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: lab3.asm

```
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
;Данные программы
DATA SEGMENT
;Директивы описания данных
a      DW 0
b      DW 0
i      DW 0
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

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

    mov cx, i
    mov ax, cx
    shl cx, 1
    mov bx, b ;
    cmp a, bx ; сравнение a и b
    jle f1second
    neg cx
    add cx, 15
    mov i1, cx
    shl cx, 1
    add cx, -33
    mov i2, cx
f1second:
    add cx, ax
    add cx, 4
    mov i1, cx
    shl cx, 1
    add cx, -18
    mov i2, cx
```

```

;расчет res
mov bx, k
cmp bx, 0
jne resSecond
    mov bx, i1
    cmp bx, i2
    jl max1
        mov cx, i2
        jmp MainFinal
    max1:
        mov cx, bx
        jmp MainFinal
resSecond:
    mov bx, i1
    cmp bx, i2
    jl max2
        mov cx, bx
        jmp MainFinal
    max2:
        mov cx, i2
        jmp MainFinal
MainFinal:                ; в cx лежит значение res
ret
Main      ENDP
CODE      ENDS
END Main

```

# ПРИЛОЖЕНИЕ Б

## ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab3.lst

Microsoft (R) Macro Assembler Version 5.10  
21:58:1

11/24/21

Page

1-1

```

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

0018          AStack  ENDS
      ;Данные программы
0000  DATA          SEGMENT
      ;Директивы описания данных
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

      ; 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

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

      ;вычисление f1
0024  8B 0E 0004 R          mov cx, i
0028  8B C1            mov ax, cx
002A  D1 E1            shl  cx, 1
002C  8B 1E 0002 R          mov bx, b      ;
0030  39 1E 0000 R          cmp  a, bx      ; сравнение a и b
0034  7E 12            jle  f1second
0036  F7 D9            neg  cx
0038  83 C1 0F          add  cx, 15
003B  89 0E 0008 R          mov i1, cx

```

```

003F D1 E1 shl cx, 1
0041 83 C1 DF add cx, -33
0044 89 0E 000A R mov i2, cx
0048 f1second:
0048 03 C8 add cx, ax
004A 83 C1 04 add cx, 4
004D 89 0E 0008 R mov i1, cx
0051 D1 E1 shl cx, 1

```

Microsoft (R) Macro Assembler Version 5.10  
21:58:1

11/24/21

Page

1-2

```

0053 83 C1 EE add cx, -18
0056 89 0E 000A R mov i2, cx

;расчет res
005A 8B 1E 0006 R mov bx, k
005E 83 FB 00 cmp bx, 0
0061 75 16 jne resSecond
0063 8B 1E 0008 R mov bx, i1
0067 3B 1E 000A R cmp bx, i2
006B 7C 07 jl max1
006D 8B 0E 000A R mov cx, i2
0071 EB 1C 90 jmp MainFinal
0074 max1:
0074 8B CB mov cx, bx
0076 EB 17 90 jmp MainFinal
0079 resSecond:
0079 8B 1E 0008 R mov bx, i1
007D 3B 1E 000A R cmp bx, i2
0081 7C 05 jl max2
0083 8B CB mov cx, bx
0085 EB 08 90 jmp MainFinal
0088 max2:
0088 8B 0E 000A R mov cx, i2
008C EB 01 90 jmp MainFinal
008F MainFinal: ; в cx лежит зна
чение res
008F CB ret
0090 Main ENDP
0090 CODE ENDS
END Main

```

Microsoft (R) Macro Assembler Version 5.10  
21:58:1

11/24/21

Symbol

s-1

# Segments and Groups:

N a m e	Length	Align	Combine Class
---------	--------	-------	---------------



```

ASTACK . . . . . 0018 PARA STACK
CODE . . . . . 0090 PARA NONE
DATA . . . . . 000C PARA NONE

```

Symbols:

	N a m e	Type	Value	Attr
	A . . . . .	L WORD	0000	DATA
	B . . . . .	L WORD	0002	DATA
	F1SECOND . . . . .	L NEAR	0048	CODE
	I . . . . .	L WORD	0004	DATA
	I1 . . . . .	L WORD	0008	DATA
	I2 . . . . .	L WORD	000A	DATA
	K . . . . .	L WORD	0006	DATA
0090	MAIN . . . . .	F PROC	0000	CODE Length =
	MAINFINAL . . . . .	L NEAR	008F	CODE
	MAX1 . . . . .	L NEAR	0074	CODE
	MAX2 . . . . .	L NEAR	0088	CODE
	RESSECOND . . . . .	L NEAR	0079	CODE
	@CPU . . . . .	TEXT	0101h	
	@FILENAME . . . . .	TEXT	LAB3	
	@VERSION . . . . .	TEXT	510	

```

80 Source Lines
80 Total Lines
20 Symbols

```

47982 + 461325 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```