

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

    ;вычисление f1
    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
    jmp f1final
f1second:
    add cx, ax
    add cx, 4
f1final:
    mov i1, cx

    ;вычисление f2
    mov cx, i
    shl cx, 1
```

```

mov ax, cx
shl cx, 1
cmp a, bx
jle f2second
    add cx, 3
    neg cx
    jmp f2final
f2second:
    add cx, ax
    add cx, -10
f2final:
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:
ret
; в cx лежит значение res

Main      ENDP
CODE      ENDS
END Main

```

ПРИЛОЖЕНИЕ Б **ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ**

Название файла: lab3.lst

Microsoft (R) Macro Assembler Version 5.10
18:42:4

11/14/21

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 0001      mov  a, 1
0012  C7 06 0002 R 0002      mov  b, 2
0018  C7 06 0004 R 0002      mov  i, 2
001E  C7 06 0006 R 0001      mov  k, 1

      ;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 08              jle  f1second
0036  F7 D9              neg  cx
0038  83 C1 0F              add  cx, 15
003B  EB 06 90              jmp  f1final

```

```

003E                                f1second:
003E 03 C8                        add cx, ax
0040 83 C1 04                    add cx, 4
0043                                f1final:
0043 89 0E 0008 R                mov i1, cx

                                ;вычисление f2
0047 8B 0E 0004 R                mov cx, i

```

Microsoft (R) Macro Assembler Version 5.10
18:42:4

11/14/21

Page

1-2

```

004B D1 E1                        shl cx, 1
004D 8B C1                        mov ax, cx
004F D1 E1                        shl cx, 1
0051 39 1E 0000 R                cmp a, bx
0055 7E 08                        jle f2second
0057 83 C1 03                    add cx, 3
005A F7 D9                        neg cx
005C EB 06 90                    jmp f2final
005F                                f2second:
005F 03 C8                        add cx, ax
0061 83 C1 F6                    add cx, -10
0064                                f2final:
0064 89 0E 000A R                mov i2, cx

                                ;рассчет res
0068 8B 1E 0006 R                mov bx, k
006C 83 FB 00                    cmp bx, 0
006F 75 16                        jne resSecond
0071 8B 1E 0008 R                mov bx, i1
0075 3B 1E 000A R                cmp bx, i2
0079 7C 07                        jl max1
007B 8B 0E 000A R                mov cx, i2
007F EB 1C 90                    jmp MainFinal
0082                                max1:
0082 8B CB                        mov cx, bx
0084 EB 17 90                    jmp MainFinal
0087                                resSecond:
0087 8B 1E 0008 R                mov bx, i1
008B 3B 1E 000A R                cmp bx, i2
008F 7C 05                        jl max2
0091 8B CB                        mov cx, bx
0093 EB 08 90                    jmp MainFinal
0096                                max2:
0096 8B 0E 000A R                mov cx, i2
009A EB 01 90                    jmp MainFinal
009D                                MainFinal:
                                ; в cx лежит зна
                                чение res
009D CB                        ret
009E                                Main      ENDP
009E                                CODE     ENDS
                                END Main

```


s-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	009E	PARA	NONE
DATA	000C	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F1FINAL	L NEAR	0043	CODE
F1SECOND	L NEAR	003E	CODE
F2FINAL	L NEAR	0064	CODE
F2SECOND	L NEAR	005F	CODE
I	L WORD	0004	DATA
I1	L WORD	0008	DATA
I2	L WORD	000A	DATA
K	L WORD	0006	DATA
009E MAIN	F PROC	0000	CODE Length =
MAINFINAL	L NEAR	009D	CODE
MAX1	L NEAR	0082	CODE
MAX2	L NEAR	0096	CODE
RESSECOND	L NEAR	0087	CODE
@CPU	TEXT	0101h	
@FILENAME	TEXT	lab3	
@VERSION	TEXT	510	

91 Source Lines
91 Total Lines
23 Symbols

47982 + 461325 Bytes symbol space free

0 Warning Errors
0 Severe Errors