

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «ОргЭВМиС**  
**Тема: Представление и обработка целых чисел. Организация**  
**ветвящихся процессов.**

Студент гр. 0382

Гудов Н.Р.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучение представления и обработки целых чисел. Изучение ветвящихся процессов.

### **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  $res = f3(i1,i2,k)$

$f1 = -(6*i - 4)$  , при  $a > b$ ;  $3*(i+2)$  , при  $a \leq b$

$f2 = -(4*i - 5)$  , при  $a > b$ ;  $10 - 3*i$  , при  $a \leq b$

$f3 = \max(i1, 10-i2)$ , при  $k < 0$ ;  $|i1 - i2|$  , при  $k \geq 0$

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  $b$ .

### **Выполнение работы.**

Первые две функции вычисляются по одинаковым условиям, поэтому выполняются одновременно. Команда *cmp* сравнивает значения  $a$  и  $b$ . Команда *jle* проверяет что  $a$  не больше  $b$ , и если условие выполняется переходит к лейблу *case\_2*. Если нет, то программа выполняется дальше до метки *cont*, позволяющей перепрыгнуть через выполнение *case\_2*. Далее значение  $k$  сравнивается с нулем. Если  $k$  не меньше 0, то переходим к лейблу *case\_02*. Внутри проверяем разность  $i1$  и  $i2$ . Для этого используется команда *jge*. Если условие выполняется переходим в метке *final*, где завершаем программу. Если же переход к *case\_02* не осуществлен, то команды выполняются последовательно до перехода к лейблу *final*.

### **Тестирование.**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	a=3 b=1 i=2 k=0	i1=-8 i2=-3 res=5	Верно Шаг 1: i1=4-6i i2=5-4i Шаг2: res= i1 - i2
2.	a=3 b=1 i=2 k= -1	i1=-8 i2=-3 res=13	Верно Шаг 1: i1=4-6i i2=5-4i Шаг2: res=max(i1,10-i2)
3.	a=1 b=3 i=2 k= -1	i1=12 i2=4 res=12	Верно Шаг 1: i1=3i+6 i2=10-3i Шаг2: res=max(i1,10-i2)
4.	a=1 b=3 i=2 k=0	i1=12 i2=4 res=8	Верно Шаг 1: i1=3i+6 i2=10-3i Шаг2: res= i1 - i2

**Выводы.**

Изучены обработка и представление целых чисел. Отработаны знания об организации ветвления на языке Ассемблера.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
AStack SEGMENT STACK
    DW 2 DUP(?)
AStack ENDS

DATA SEGMENT
    a    DW 1
    b    DW 3
    i    DW 2
    k    DW 0
    i1   DW ?
    i2   DW ?
    res  DW ?
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
Main PROC FAR
    push DS
    sub ax, ax
    push ax
    mov ax, DATA
    mov DS, ax

    mov ax, i
    mov bx, i
    shl ax, 1
    shl ax, 1
    shl ax, 1    ;ax=8i
    shl bx, 1    ;bx=2i
    sub ax, bx   ;ax=6i

    mov cx, a
    cmp cx, b
    jle case_2

    mov i1, 4    ;i1=4
    sub i1, ax   ;i1=4-6i

    mov ax, i1   ;ax=4-6i
    add ax, bx   ;ax=4-4i
    add ax, 1    ;ax=5-4i
    mov i2, ax   ;i2=5-4i

    jmp cont

case_2:
```

```

        shr ax, 1  ;ax=3i
        mov i1, ax ;i1=3i
        add i1, 6  ;i1=3i+6

        mov i2, 10 ;i2=10
        sub i2, ax  ;i2=10-3i

cont:

        mov ax, i1
        mov bx, 10
        sub bx, i2 ;bx=10-i2

        mov cx, k
        cmp k, 0
        jge case_02

        mov res, ax
        cmp ax, bx
        jge final
        mov res, bx
        jmp final

case_02:

        sub ax, i2 ;ax=i1-i2
        mov res, ax
        cmp ax, 0
        jge final

        neg ax
        mov res, ax

final:
        ret
Main ENDP
CODE ENDS
END Main

```

# ПРИЛОЖЕНИЕ Б

## ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Microsoft (R) Macro Assembler Version 5.10

11/1/21

19:00:18

Page

1-1

```
0000          AStack SEGMENT STACK
0000 0002[          DW 2 DUP(?)
          ????
          ]

0004          AStack ENDS

0000          DATA SEGMENT
0000 0001          a    DW 1
0002 0003          b    DW 3
0004 0002          i    DW 2
0006 0000          k    DW 0
0008 0000          i1   DW ?
000A 0000          i2   DW ?
000C 0000          res  DW ?
000E          DATA ENDS

0000          CODE SEGMENT

          ASSUME CS:CODE, DS:DATA, SS:AStack

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 A1 0004 R      mov ax, i
000C 8B 1E 0004 R      mov bx, i
0010 D1 E0          shl ax, 1
0012 D1 E0          shl ax, 1
```

```

0014 D1 E0          shl ax, 1  ;ax=8i
0016 D1 E3          shl bx, 1  ;bx=2i
0018 2B C3          sub ax, bx ;ax=6i

001A 8B 0E 0000 R    mov cx, a
001E 3B 0E 0002 R    cmp cx, b
0022 7E 18          jle case_2

0024 C7 06 0008 R 0004    mov i1, 4  ;i1=4
002A 29 06 0008 R          sub i1, ax ;i1=4-6i

002E A1 0008 R        mov ax, i1 ;ax=4-6i
0031 03 C3          add ax, bx ;ax=4-4i
0033 05 0001        add ax, 1  ;ax=5-4i
0036 A3 000A R      mov i2, ax ;i2=5-4i

0039 EB 15 90          jmp cont

003C                case_2:

003C D1 E8          shr ax, 1  ;ax=3i
003E A3 0008 R      mov i1, ax ;i1=3i
0041 83 06 0008 R 06    add i1, 6  ;i1=3i+6

```

Microsoft (R) Macro Assembler Version 5.10

11/1/21

19:00:18

Page

1-2

```

0046 C7 06 000A R 000A    mov i2, 10  ;i2=10
004C 29 06 000A R          sub i2, ax  ;i2=10-3i

0050                cont:

0050 A1 0008 R        mov ax, i1
0053 BB 000A          mov bx, 10
0056 2B 1E 000A R      sub bx, i2 ;bx=10-i2

```



```

005A  8B 0E 0006 R          mov cx, k
005E  83 3E 0006 R 00      cmp k, 0
0063  7D 0E                jge case_02

0065  A3 000C R          mov res, ax
0068  3B C3                cmp ax, bx
006A  7D 18                jge final
006C  89 1E 000C R          mov res, bx
0070  EB 12 90            jmp final

0073                                case_02:

0073  2B 06 000A R          sub ax, i2 ;ax=i1-i2
0077  A3 000C R          mov res, ax
007A  3D 0000                cmp ax, 0
007D  7D 05                jge final

007F  F7 D8                neg ax
0081  A3 000C R          mov res, ax

0084                                final:
0084  CB                    ret
0085                                Main ENDP
0085                                CODE ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10

11/1/21

19:00:18

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK	. . . . .	0004	PARA	STACK
CODE	. . . . .	0085	PARA	NONE
DATA	. . . . .	000E	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
A	. . . . .	L WORD	0000	DATA
B	. . . . .	L WORD	0002	DATA
CASE_02	. . . . .	L NEAR	0073	CODE
CASE_2	. . . . .	L NEAR	003C	CODE
CONT	. . . . .	L NEAR	0050	CODE
FINAL	. . . . .	L NEAR	0084	CODE
I	. . . . .	L WORD	0004	DATA
I1	. . . . .	L WORD	0008	DATA
I2	. . . . .	L WORD	000A	DATA
K	. . . . .	L WORD	0006	DATA
MAIN	. . . . .	F PROC	0000	CODE Length =
0085				
RES	. . . . .	L WORD	000C	DATA
@CPU	. . . . .	TEXT	0101h	
@FILENAME	. . . . .	TEXT	lab3	
@VERSION	. . . . .	TEXT	510	

85 Source Lines  
85 Total Lines  
20 Symbols

48056 + 461251 Bytes symbol space free

0 Warning Errors  
0 Severe Errors