

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. 0382

Осинкин Е. А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучение представление и обработку целых чисел, также научиться организовывать ветвящиеся процессы на языке Ассемблера.

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 .

Выполнение работы.

Вариант 26.

$i1$:

$$f6 = \begin{cases} / 2*(i+1) - 4, & \text{при } a > b \\ \backslash 5 - 3*(i+1), & \text{при } a \leq b \end{cases}$$

$i2$:

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

res :

$$f1 = \begin{cases} / \min(i1,i2), & \text{при } k=0 \\ \backslash \max(i1,i2), & \text{при } k \neq 0 \end{cases}$$

Сначала определяются все сегменты программы, сегмент стека AStack, сегмент данных DATA SEGMENT, в котором объявляются переменные a,b,i,k,i1,i2,res и сегмент кода CODE SEGMENT. Выполнение программы начинается с процедуры MAIN, в которой первым делом задаётся сегмент данных, далее сравниваются два числа a и b, и в зависимости от результата вычисляются значения i1 и i2, переход к нужному случаю происходит посредством условного перехода jle(если $a \leq b$), далее к метке Label_res переход осуществляется либо естественным образом, либо при помощи безусловного перехода jmp. Далее проверяется равенство k нулю, и в зависимости от результата определяется результат res, при этом используются условные переходы jne($k \neq 0$), jge($i1 \geq i2$).

Исходный программный код смотрите в приложении А. Файл листинга смотрите в приложении Б.

Тестирование.

Результаты представлены в таблице 1.

Таблица 1 – Результаты тестирования

№	Входные данные	Выходные данные	Комментарий
1	a=5 b=5 i=-2 k=0	i1=8 i2=18 res=8	Правильно. $5-3*(-2+1)=8$ $9-3*(-2-1)=18$ $\min(8,18)=8$
2	a=6 b=5 i=-2 k=1	i1=-6(FFFA) i2=4 res=4	Правильно. $2*(-2+1)-4=-6$ $-(6*(-2)+8)=4$ $\max(-6,4)=4$
3	a=5 b=5 i=-1	i1=5 i2=15 res=15	Правильно. $5-3*(-1+1)=5$ $9-3*(-1-1)=15$

	$k=1$		$\max(5,15)=15$
--	-------	--	-----------------

Выводы.

В ходе работы были изучены способы ветвления программы, условные и безусловные переходы, также написана программа, вычисляющая значение функции по заданным целочисленным параметрам.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab.asm

```
Astack SEGMENT STACK
```

```
    DW 12 DUP(?)
```

```
Astack ENDS
```

```
DATA SEGMENT
```

```
    a DW 6
```

```
    b DW 5
```

```
    i DW -2
```

```
    k DW 1
```

```
    i1 DW 0 ;f1=2*(i+1)-4 if a>b, if a<=b f1=5-3*(i+1) f1=2i-2 f1=-3i+2
```

```
    i2 DW 0 ;f2=-(6*i+8) if a>b, if a<=bx f2=9-3*(i-1) f2=-6i-8 f2=12-3i
```

```
    res DW 0 ;f3=min(i1,i2) if k=0, if k!=0 f3=max(i1,i2)
```

```
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 cx,i ;cx=i
```

```
    shl cx,1 ;cx=2i
```

```
    mov ax,a
```

```
    cmp ax,b
```

```
    jle Label_f1_2
```

```
Label_f1_1:
```

```
    sub cx,2 ;cx=2i-2
```

```
    mov i1,cx
```

```
    shl cx,1 ;cx=4i-4
```

```
    add cx,i ;cx=5i-4
```

```
    add cx,i ;cx=6i-4
```

```
    add cx,12 ;cx=6i+8
```

```
    neg cx ;cx=-(6i+8)
```

```
    mov i2,cx
```

```
    jmp Label_res
```

```
Label_f1_2:
```

```
    sub cx, 2 ;cx=2i-2
```

```
    add cx,i ;cx=3i-2
```

```
    neg cx ;cx=2-3i
```

```
    mov i1,cx
```

```

    add cx, 10    ;cx=12-3i
    mov i2,cx

Label_res:      ;f3=min(i1,i2) if k=0, if k!=0 f3=max(i1,i2)
    mov bx,i1    ;bx=i1
    mov res,cx   ;res=i2
    cmp k,0
    jne Label_res_else ;if k!=0
    cmp bx,i2    ;i1?i2
    jge final
    mov res,bx
    jmp final

Label_res_else:
    cmp bx,i2    ;i1?i2
    jle final
    mov res,bx
    jmp final

final:
    ret
Main ENDP
CODE ENDS
    END Main

```

ПРИЛОЖЕНИЕ Б

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab.lst

Microsoft (R) Macro Assembler Version 5.10
01:45:1

11/25/21

Page

1-1

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

0018                                AStack ENDS

0000                                DATA SEGMENT

0000 0006                            a DW 6
0002 0005                            b DW 5
0004 FFFE                            i DW -2
0006 0001                            k DW 1
0008 0000                            i1 DW 0 ;f1=2*(i+1)-4 if a>b, if a<=b
f1=5
                                -3*(i+1) f1=2i-2 f1=-3i+2
000A 0000                            i2 DW 0 ;f2=-(6*i+8) if a>b, if a<=bx
f2=9
                                -3*(i-1) f2=-6i-8 f2=12-3i
000C 0000                            res DW 0 ;f3=min(i1,i2) if k=0, if
k!=0 f3=
                                max(i1,i2)

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  8B 0E 0004 R                mov cx,i      ;cx=i
000D  D1 E1                      shl cx,1      ;cx=2i

000F  A1 0000 R                  mov ax,a
0012  3B 06 0002 R                cmp ax,b
0016  7E 1D                      jle Label_fl_2

0018                          Label_fl_1:
0018  83 E9 02                      sub cx,2      ;cx=2i-2
001B  89 0E 0008 R                mov i1,cx

001F  D1 E1                      shl cx,1      ;cx=4i-4
0021  03 0E 0004 R                add cx,i      ;cx=5i-4
0025  03 0E 0004 R                add cx,i      ;cx=6i-4
0029  83 C1 0C                      add cx,12     ;cx=6i+8
002C  F7 D9                      neg cx        ;cx=-(6i+8)
002E  89 0E 000A R                mov i2,cx
0032  EB 15 90                      jmp Label_res

0035                          Label_fl_2:
0035  83 E9 02                      sub cx, 2     ;cx=2i-2

Microsoft (R) Macro Assembler Version 5.10      11/25/21
01:45:1

1-2

0038  03 0E 0004 R                add cx,i      ;cx=3i-2
003C  F7 D9                      neg cx        ;cx=2-3i
003E  89 0E 0008 R                mov i1,cx

0042  83 C1 0A                      add cx, 10    ;cx=12-3i
0045  89 0E 000A R                mov i2,cx

0049                          Label_res:      ;f3=min(i1,i2) if k=0, if

```


k!=0 f3

=max(i1,i2)

```
0049  8B 1E 0008 R      mov bx,i1    ;bx=i1
004D  89 0E 000C R      mov res,cx   ;res=i2
0051  83 3E 0006 R 00      cmp k,0
0056  75 0D              jne Label_res_else ;if k!=0
0058  3B 1E 000A R      cmp bx,i2    ;i1?i2
005C  7D 14              jge final
005E  89 1E 000C R      mov res,bx
0062  EB 0E 90              jmp final
```

```
0065                      Label_res_else:
0065  3B 1E 000A R      cmp bx,i2    ;i1?i2
0069  7E 07              jle final
006B  89 1E 000C R      mov res,bx
006F  EB 01 90              jmp final
```

```
0072                      final:
0072  CB                  ret
0073                      Main ENDP
0073                      CODE ENDS
```

END Main

Microsoft (R) Macro Assembler Version 5.10
01:45:1

11/25/21

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	0018	PARA	STACK
CODE	0073	PARA	NONE
DATA	000E	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
---------	------	-------	------

A	L WORD	0000 DATA	
B	L WORD	0002 DATA	
FINAL	L NEAR	0072 CODE	
I	L WORD	0004 DATA	
I1	L WORD	0008 DATA	
I2	L WORD	000A DATA	
K	L WORD	0006 DATA	
LABEL_F1_1	L NEAR	0018 CODE	
LABEL_F1_2	L NEAR	0035 CODE	
LABEL_RES	L NEAR	0049 CODE	
LABEL_RES_ELSE	L NEAR	0065 CODE	
MAIN	F PROC	0000 CODE	Length = 0073
RES	L WORD	000C DATA	
@CPU	TEXT	0101h	
@FILENAME	TEXT	lab	
@VERSION	TEXT	510	

76 Source Lines

76 Total Lines

21 Symbols

48030 + 461277 Bytes symbol space free

0 Warning Errors

0 Severe Errors