

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

Сергеев Д.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 5
```

```
    b DW 5
```

```
    i DW -1
```

```
    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
```

```
    mov dx,cx ;dx=i
```

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

```
    mov di,cx ;di=2i
```

```
    mov ax,a
```

```
    cmp ax,b
```

```
    jle Label_f1_2
```

```
Label_f1_1:
```

```
    mov ax,di ;ax=2i
```

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

```
    mov i1,ax
```

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

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

```
    add cx,di ;cx=6i
```

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

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

```
    mov i2,cx
```

```
    jmp Label_res
```

```
Label_f1_2:
```

```
    mov ax,2 ;ax=2
```

```
    sub ax,di ;ax=2-2i
```

```
    sub ax,dx ;ax=2-3i
```

```
    mov i1,ax
```

```

    add cx,dx    ;cx=3i
    neg cx       ;cx=-3i
    mov ax,12
    add ax,cx
    mov i2,ax

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

Label_res_else:
    cmp ax,i2
    jge final
    mov res,bx
    jmp final

final:

Main ENDP
CODE ENDS
    END Main

```

ПРИЛОЖЕНИЕ Б

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab.lst

Microsoft (R) Macro Assembler Version 5.10

11/1/21

17:37:22

Page

1-1

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

0018          AStack ENDS

0000          DATA SEGMENT

0000 0005          a DW 5
0002 0005          b DW 5
0004 FFFF          i DW -1
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 8B D1                            mov dx,cx      ;dx=i
000F D1 E1                            shl cx,1      ;cx=2i
0011 8B F9                            mov di,cx      ;di=2i

0013 A1 0000 R                        mov ax,a
0016 3B 06 0002 R                    cmp ax,b
001A 7E 18                            jle Label_f1_2

001C                                Label_f1_1:
001C 8B C7                            mov ax,di      ;ax=2i
001E 2D 0002                            sub ax,2      ;ax=2i-2
0021 A3 0008 R                        mov i1,ax
                                ;f2=-(6*i+8) if a>b, if a<=bx
f
                                2=9-3*(i-1)
0024 D1 E1                            shl cx,1      ;cx=4i
0026 03 CF                            add cx,di      ;cx=6i
0028 83 C1 08                            add cx,8      ;cx=6i+8
002B F7 D9                            neg cx        ;cx=-(6i+8)
002D 89 0E 000A R                    mov i2,cx
0031 EB 17 90                            jmp Label_res

```

Microsoft (R) Macro Assembler Version 5.10

11/1/21

17:37:22

Page

1-2

```

0034                                Label_f1_2:
0034 B8 0002                            mov ax,2      ;ax=2
0037 2B C7                            sub ax, di    ;ax=2-2i

```



```

0039  2B C2                      sub ax, dx  ;ax=2-3i
003B  A3 0008 R                  mov i1,ax

003E  03 CA                      add cx,dx  ;cx=3i
0040  F7 D9                      neg cx    ;cx=-3i
0042  B8 000C                    mov ax,12
0045  03 C1                      add ax,cx
0047  A3 000A R                  mov i2,ax

004A                      Label_res:      ;f3=min(i1,i2) if k=0, if k!=0 f3
                                =max(i1,i2)
004A  8B 1E 000A R                  mov bx,i2
004E  A3 000C R                  mov res,ax
0051  83 3E 0006 R 00              cmp k,0
0056  75 0D                      jne Label_res_else
0058  3B 06 000A R                  cmp ax,i2
005C  7E 14                      jle final
005E  89 1E 000C R                  mov res,bx
0062  EB 0E 90                      jmp final

0065                      Label_res_else:
0065  3B 06 000A R                  cmp ax,i2
0069  7D 07                      jge final
006B  89 1E 000C R                  mov res,bx
006F  EB 01 90                      jmp final

0072                      final:

0072                      Main ENDP
0072                      CODE ENDS

                                END Main

```

Microsoft (R) Macro Assembler Version 5.10

11/1/21

17:37:22

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0072	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	001C	CODE
LABEL_F1_2	L NEAR	0034	CODE
LABEL_RES	L NEAR	004A	CODE
LABEL_RES_ELSE	L NEAR	0065	CODE
MAIN	F PROC	0000	CODE Length =
RES	L WORD	000C	DATA
@CPU	TEXT	0101h	
@FILENAME	TEXT	LAB	
@VERSION	TEXT	510	

0072

81 Source Lines
81 Total Lines

21 Symbols

48030 + 461277 Bytes symbol space free

0 Warning Errors

0 Severe Errors