МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Организация ЭВМ и систем»

Тема: Представление и обработка целых чисел. Организация ветвящихся процессов

Студент гр. 9383 _____ Арутюнян С.Н.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Содержание

1. Цель работы……3
2. Текст программы lab3.asm……5
3. Текст листинга lab3.lst……9
Выводы……..15

1. Цель работы

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет:

a) значения функций i1 = f1(a,b,i) и i2 = f2(a,b,i);

b) 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.

Вид функций:

f1 = 15 — 2i при a > b; 3i + 4 при a <= b

f2 = -(4i + 3) при a > b; 6i — 10 при a <= b

f3 = min(i1, i2) при k = 0; max(i1, i2) при k != 0

2. Текст программы lab3.asm

```
; Стек программы
AStack SEGMENT STACK
   DW 12 DUP(?)
AStack ENDS

; Данные программы
DATA SEGMENT
a DW 1
b DW 2
i DW 3
k DW 4
i1 DW ?
i2 DW ?
result DW ?
DATA ENDS


; Код программы
CODE SEGMENT
   ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
Main PROC FAR
   mov ax, DATA
   mov ds, ax

f1:
   mov ax, a
```

```
    mov bx, b
    cmp ax, bx

    jg f1_great ; if ax > bx

; 3i + 4
f1_less:
    ; 3*i
    mov dx, i
    ; *2
    shl dx, 1
    ; +i
    add dx, i
    ; +4
    add dx, 4

    mov i1, dx
    jmp f2

; 15 - 2*i
f1_great:
    ; 2*i
    mov dx, i
    shl dx, 1
    ; -15
    sub dx, 15
    neg dx

    mov i1, dx
```

```
f2:
    mov ax, a
    mov bx, b
    cmp ax, bx

    jg f2_great

; 6i - 10
f2_less:
    mov dx, i
    ; *4
    mov cl, 2
    shl dx, cl
    ; +i
    add dx, i
    ; +i
    add dx, i
    ; -10
    sub dx, 10

    mov i2, dx
    jmp res

; -(4i + 3)
f2_great:
    mov dx, i
    ; *4
```

```
    mov cl, 2
    shl dx, cl
    ; +3
    add dx, 3
    neg dx

    mov i2, dx



; f3
res:
    mov ax, i1
    mov bx, i2
    mov cx, k

    ; упорядочиваем ax и bx
    cmp ax, bx
    jg min
    jmp res2

min:
    xchg ax, bx

res2:
    cmp k, 0
    jz res_zero

; k != 0
res_nonzero:
```

```
    ; максимальный в bx
    mov result, bx
    jmp end_prog

; k = 0
res_zero:
    ; минимальный в ax
    mov result, ax

end_prog:
    mov ah, 4ch
    int 21h

Main ENDP
CODE ENDS
    END Main
```

3. Текст листинга lab3.lst

```
#Microsoft (R) Macro Assembler Version 5.10                10/29/20 10:01:4
                                                            Page     1-1
                                ; Стек программы
0000                            AStack SEGMENT STACK
0000  000C[                             DW 12 DUP(?)
        ????
                              ]

0018                            AStack ENDS

                                ; Данные программы
0000                            DATA SEGMENT
0000  0001                      a DW 1
0002  0002                      b DW 2
0004  0003                      i DW 3
0006  0004                      k DW 4
0008  0000                      i1 DW ?
000A  0000                      i2 DW ?
000C  0000                      result DW ?
000E                            DATA ENDS
                                ; Код программы
0000                            CODE SEGMENT
                                   ASSUME CS:CODE, DS:DATA, SS:AStack

                                ; Головная процедура
0000                            Main PROC FAR
0000  B8 ---- R                    mov ax, DATA
0003  8E D8                               mov ds, ax
```

```
0005                    f1:
0005  A1 0000 R            mov ax, a
0008  8B 1E 0002 R              mov bx, b
000C  3B C3                    cmp ax, bx

000E  7F 14                    jg f1_great ; if ax > bx

                        ; 3i + 4
0010                    f1_less:
                           ; 3*i
0010  8B 16 0004 R             mov dx, i
                           ; *2
0014  D1 E2                    shl dx, 1
                           ; +i
0016  03 16 0004 R             add dx, i
                           ; +4
001A  83 C2 04                 add dx, 4

001D  89 16 0008 R             mov i1, dx
0021  EB 10 90                 jmp f2

                        ; 15 - 2*i
0024                    f1_great:
                           ; 2*i
0024  8B 16 0004 R             mov dx, i
#Microsoft (R) Macro Assembler Version 5.10          10/29/20 10:01:4
                                    Page     1-2
0028  D1 E2                    shl dx, 1
```

```
                        ; -15
002A  83 EA 0F              sub dx, 15
002D  F7 DA                 neg dx

002F  89 16 0008 R          mov i1, dx
0033                    f2:
0033  A1 0000 R         mov ax, a
0036  8B 1E 0002 R          mov bx, b
003A  3B C3                 cmp ax, bx

003C  7F 1A                 jg f2_great

                        ; 6i - 10
003E                    f2_less:
003E  8B 16 0004 R          mov dx, i
                        ; *4
0042  B1 02                 mov cl, 2
0044  D3 E2                 shl dx, cl
                        ; +i
0046  03 16 0004 R          add dx, i
                        ; +i
004A  03 16 0004 R          add dx, i
                        ; -10
004E  83 EA 0A              sub dx, 10

0051  89 16 000A R          mov i2, dx
0055  EB 12 90              jmp res

                        ; -(4i + 3)
```

```
0058                    f2_great:
0058  8B 16 0004 R              mov dx, i
                        ; *4
005C  B1 02                     mov cl, 2
005E  D3 E2                     shl dx, cl
                        ; +3
0060  83 C2 03                  add dx, 3
0063  F7 DA                     neg dx

0065  89 16 000A R              mov i2, dx
                    ; f3
0069                    res:
0069  A1 0008 R             mov ax, i1
006C  8B 1E 000A R              mov bx, i2
0070  8B 0E 0006 R              mov cx, k

                        ; упорядочиваем ax и bx
0074  3B C3                     cmp ax, bx
0076  7F 03                 jg min
0078  EB 02 90                  jmp res2
#Microsoft (R) Macro Assembler Version 5.10          10/29/20 10:01:4
                                    Page     1-3
007B                    min:
007B  93                    xchg ax, bx

007C                    res2:
007C  83 3E 0006 R 00           cmp k, 0
0081  74 07                 jz res_zero
```

```
                        ; k != 0
0083                    res_nonzero:
                            ; максимальный в bx
0083  89 1E 000C R               mov result, bx
0087  EB 04 90                   jmp end_prog

                        ; k = 0
008A                    res_zero:
                            ; минимальный в ax
008A  A3 000C R             mov result, ax
008D                    end_prog:
008D  B4 4C                      mov ah, 4ch
008F  CD 21                      int 21h

0091                    Main ENDP
0091                    CODE ENDS
                            END Main
#Microsoft (R) Macro Assembler Version 5.10          10/29/20 10:01:4
                                        Symbols-1
Segments and Groups:
                N a m e         Length     Align   Combine Class
ASTACK . . . . . . . . . . . .  0018  PARA      STACK
CODE . . . . . . . . . . . . .  0091  PARA      NONE
DATA . . . . . . . . . . . . .  000E  PARA      NONE
Symbols:
                N a m e         Type    Value   Attr
A  . . . . . . . . . . . . . .  L WORD   0000  DATA
B  . . . . . . . . . . . . . .  L WORD   0002  DATA
END_PROG . . . . . . . . . . .  L NEAR   008D  CODE
```

```
F1 . . . . . .  L NEAR  0005  CODE
F1_GREAT . . . . . .  L NEAR  0024  CODE
F1_LESS . . . . . .  L NEAR  0010  CODE
F2 . . . . . .  L NEAR  0033  CODE
F2_GREAT . . . . . .  L NEAR  0058  CODE
F2_LESS . . . . . .  L NEAR  003E  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 = 0091
MIN . . . . . .  L NEAR  007B  CODE
RES . . . . . .  L NEAR  0069  CODE
RES2 . . . . . .  L NEAR  007C  CODE
RESULT . . . . . .  L WORD  000C  DATA
RES_NONZERO . . . . . .  L NEAR  0083  CODE
RES_ZERO . . . . . .  L NEAR  008A  CODE
@CPU . . . . . .  TEXT  0101h
@FILENAME . . . . . .  TEXT  lab2
@VERSION . . . . . .  TEXT  510
#Microsoft (R) Macro Assembler Version 5.10  10/29/20 10:01:4
                                             Symbols-2
   131 Source  Lines
   131 Total   Lines
    28 Symbols
 48040 + 457170 Bytes symbol space free
     0 Warning Errors
     0 Severe  Errors
```

Выводы

В процессе выполнения лабораторной работы было изучено представление и обработка целых чисел, а также получены навыки работы с организацией ветвящихся процессов.