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. 0383 _____ Козлов Т.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

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.

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.

Ход работы.

Вариант 4 (1.5.4):

```
        / 15-2*i , при a>b
 f1 = <
        \ 3*i+4 , при a<=b
       /20 - 4*i , при a>b
f5 = <
       \ -(6*I - 6), при a<=b
       / min (|i1 - i2|, 2), при k<0
f4 = <
        \ max( -6, -i2), при k>=0
```

Для сравнения использовалась команды cmd (выполняющий сравнение и в результате изменяя флаги) и команды jle, jge, jl, работающие с положительными и отрицательными числами, проверяющие флаги, в которые внесла изменения команда cmd.

Команды jle, jge, jl выполняли (или не выполняли) короткий переход на метки, в результате чего выполнялись именно те команды, которые нужны для вычисления значения функции при данных условиях.

Табл.1: Тестирование работы lab3.asm

Значения a, b, i, k	Результат работы f1 (в i1)	Результат работы f5 (в i2)	Результат работы f4 (в CX)	Комментарий
a = 3 b = 5 i = 8 k = 13	001C = 28	FFBE = -42	002A = 42	Верно
a = 3 b = 5 i = 8 k = -1	001C = 28	FFBE = -42	0002 = 2	Верно
a = 5 b = 3 i = 8 k = 2	FFFF = -1	FFEE = -12	000C = 12	Верно
a = 5 b = 3 i = 8 k = -2	FFFF = -1	FFEE = -12	0002 = 2	Верно

Компоненты программы см. в приложении А.

Выводы.

В ходе выполнения работы были изучены способы работы с целыми положительными и отрицательными числами, изучены условные переходы и реализованы сравнения на языке Ассемблер.

ПРИЛОЖЕНИЕ А

Тексты компонентов программы lab3.exe

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    13
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, 5
mov b, 3
mov i, 8
mov k, -2

;f1&f2
mov cx, i
shl cx, 1 ; cx = 2*i
mov ax, cx ; ax = 2*i
mov bx, b

cmp a, bx ; сравнение a и b
jle f1f2
        neg cx ; a > b
        add cx, 15
        mov i1, cx

        mov i2, cx
        sub i2, ax
        add i2, 5
        jmp f1f2End

f1f2: ; a <= b
        add cx, i ; cx = 3i
        mov i1, cx
        add i1, 4

        neg cx
        add cx, cx
```

```
        add cx, 6
        mov i2, cx
f1f2End:

;f4
mov cx, i1
sub cx, i2
cmp cx, 0
jge module
  neg cx
module: ; cx = |i1 - i2|

mov bx, k
cmp bx, 0 ; сравнение k и 0
jge f4
  mov bx, 2 ; k < 0
        cmp cx, bx
        jl min
          mov cx, 2
          jmp MainEnd
        min:
          jmp MainEnd
f4:         ; k >= 0
mov cx, i2
neg cx
mov bx, -6
cmp bx, cx
jge max
        mov cx, bx
max:
```

```
        MainEnd: ; в cx лежит значение функции f4
    ret
Main    ENDP
CODE    ENDS
END Main
```

Lab3.lst:

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

0018                    AStack  ENDS
                        ;P”P°PSPSC‹Pµ PїCЂPsPiCЂP°PjPjC‹
0000                    DATA    SEGMENT
                        ;P”PёCЂPµPєC,PёPIC‹ PsPїPёCЃP°PSPёCЏ PґP°PSPSC‹
                        C…
0000  0000              a     DW    0
0002  0000              b     DW    0
0004  0000              i     DW    0
0006  000D              k     DW    13
0008  0000              i1    DW    0
000A  0000              i2    DW    0

000C                    DATA    ENDS

                        ; PљPsPґ PїCЂPsPiCЂP°PjPjC‹
0000                    CODE    SEGMENT
                            ASSUME CS:CODE, DS:DATA, SS:AStack

                        ; P“PsP»PsPIPSP°CЏ PїCЂPsC†PµPґCґCЂP°
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 0005                 mov a, 5
0012  C7 06 0002 R 0003                 mov b, 3
0018  C7 06 0004 R 0008                 mov i, 8
001E  C7 06 0006 R FFFE                 mov k, -2

                                ;f1&f2
0024  8B 0E 0004 R                      mov cx, i
0028  D1 E1                             shl cx, 1 ; cx = 2*i
002A  8B C1                             mov ax, cx ; ax = 2*i
002C  8B 1E 0002 R                      mov bx, b

0030  39 1E 0000 R                      cmp a, bx ; CЃCЂP°PIPSPµPSPëPµ a Pë b
0034  7E 19                             jle f1f2
0036  F7 D9                                 neg cx ; a > b
0038  83 C1 0F                              add cx, 15
003B  89 0E 0008 R                          mov i1, cx

003F  89 0E 000A R                          mov i2, cx
0043  29 06 000A R                          sub i2, ax
0047  83 06 000A R 05                       add i2, 5
004C  EB 19 90                              jmp f1f2End
```

```
004F                    f1f2: ; a <= b
004F  03 0E 0004 R              add cx, i ; cx = 3i
0053  89 0E 0008 R              mov i1, cx
0057  83 06 0008 R 04           add i1, 4

005C  F7 D9                     neg cx
005E  03 C9                     add cx, cx
0060  83 C1 06                  add cx, 6
0063  89 0E 000A R              mov i2, cx
0067                    f1f2End:

                        ;f4
0067  8B 0E 0008 R          mov cx, i1
006B  2B 0E 000A R          sub cx, i2
006F  83 F9 00              cmp cx, 0
0072  7D 02                 jge module
0074  F7 D9                   neg cx
0076                    module: ; cx = |i1 - i2|

0076  8B 1E 0006 R          mov bx, k
007A  83 FB 00              cmp bx, 0 ; CЃCЂP°PIPSPµPSPëPµ k Pë 0
007D  7D 10                 jge f4
007F  BB 0002                 mov bx, 2 ; k < 0
0082  3B CB                     cmp cx, bx
0084  7C 06                     jl min
0086  B9 0002                     mov cx, 2
```

```
0089  EB 13 90                          jmp MainEnd
008C                                min:
008C  EB 10 90                          jmp MainEnd
008F                              f4:       ; k >= 0
008F  8B 0E 000A R                        mov cx, i2
0093  F7 D9                               neg cx
0095  BB FFFA                             mov bx, -6
0098  3B D9                               cmp bx, cx
009A  7D 02                               jge max
009C  8B CB                                 mov cx, bx
009E                                  max:
009E                                  MainEnd: ; PI cx P»PµP¶PëC, P·PSP°C‡P
                                  µPSPëPµ C„CŕPSPєC†PëPë f4
009E  CB                            ret
009F                              Main      ENDP
009F                              CODE      ENDS
                                  END Main
```

Segments and Groups:

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

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F1F2	L NEAR	004F	CODE
F1F2END	L NEAR	0067	CODE
F4	L NEAR	008F	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 = 009F

```
MAINEND . . . . . .     L NEAR     009E  CODE
MAX . . . . . .  L NEAR     009E  CODE
MIN . . . . . .  L NEAR     008C  CODE
MODULE . . . . . .      L NEAR     0076  CODE

@CPU . . . . . .        TEXT  0101h
@FILENAME  . . . . . .  TEXT  lab3
@VERSION . . . . . .    TEXT  510


    92 Source  Lines
    92 Total   Lines
    22 Symbols

 47978 + 461329 Bytes symbol space free

     0 Warning Errors
     0 Severe  Errors
```