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. _____ Кирсанов А.Я.

Преподаватель _____ Кирьянчиков В.А.

Санкт-Петербург
2017

Задание.

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),

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.

Вариант 8(2,3,8):

```
      / -(4*i+3), при  a>b
f2 = <
      \ 6*i-10 , при  a<=b


      / 7-4*i ,  при  a>b
f3 = <
      \ 8-6*i , при  a<=b


      / (|i2|-|i1|),  при  k<0
f8 = <
      \ max(4,|i2|-3), при  k>=0
```

Описание программы.

Ветвление организуется с помощью 2 меток и одного сравнения, имитируя конструкцию if-else высокоуровневого ЯП: после сравнения может использоваться метка для перехода на блок «else», а может выполняться список команд блока «if» сразу после сравнения, в конце которого находится переход на метку, расположенную после блока «else» (для такой метки используется суффикс e - exit). Для умножения числа на 2 используется побитовый сдвиг числа влево на одну позицию.

Тестирование.

Таблица 1. Тестирование.

№	a	b	i	k	i1	i2	Res
1	1	2	5	1	0014=20	FFEA=-22	0013=19
2	1	2	5	0	0014=20	FFEA=-22	0013=19
3	2	1	5	0	FFE9=-23	FFF3=-13	000A=10
4	2	1	3	-1	FFF1=-15	FFFB=-5	0014=20
5	-1	2	5	1	0014=20	FFEA=-22	0013=19
6	-1	-2	2	0	FFF5=-11	FFFF=-1	0004=4

Результаты тестирования соответствуют правильной работе программы.

Вывод.

В процессе работы были изучены представление и обработка целых чисел, организация ветвящихся процессов и разработана и протестирована программа, вычисляющая значение 3 функций.

Приложение.

Исходный файл программы:

```
; Учебная программа  лабораторной работы №3 по дисциплине "Архитектура компьютера"
;
; Стек  программы
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    0
i1      DW    0
i2      DW    0
res     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 ax,i
             cmp ax,b
             jle l1
             ;a>b
                   ;первая функция
                   sal ax,1
                   sal ax,1
                   neg ax
                   mov cx,ax
                   sub ax,3
                   mov i1,ax
                   ;вторая функция
                   add cx,7
                   mov i2,cx
                   jmp l1e
             l1:
```

```
;a<=b
       ;первая функция
    sal ax,1
       sal ax,1
       add ax,i
       add ax,i
       neg ax
       mov cx,ax
       add ax,10
       neg ax
       mov i1,ax
       ;вторая функция
       add cx,8
       mov i2, cx
l1e:

cmp k,0 ; третья функция
jl l2
;k>=0
       mov ax,i2
       cmp ax,0
       jns ns
              neg ax ;если i2<0
       ns:
       sub ax,3
       cmp ax,4
       jg max ; ax>4 - переходим на метку max
              mov ax,4 ; иначе кладем в ax 4
       max:
       jmp l2e
l2:
;k<0
       mov ax, i2
       cmp i2,0
       jns gi2
              neg ax ;если i2<0
       gi2:
       cmp i1,0
       jns gi1
              neg ax ;если i1<0 делаем i2<0
       gi1:
       add ax,i1
       cmp ax,0
       jns sum
              neg ax ;если i1+i2<0
       sum:
       mov res,ax
```

```
            l2e:

            mov res,ax ;кладем значение третьей функции
            ret
Main    ENDP
CODE    ENDS
        END Main
```

Листинг программы:

```
Microsoft (R) Macro Assembler Version 5.10                  11/7/18 06:28:06
                                                             Page     1-1
                          ; Учебная программа  лабор
                          аторной работы №3 по дисцЍ
                          ◌ِسبِ□▯"Архитектура компьэ
                           บฺ͓PT
                          ;
                          ; Стек  программы

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

 0018                     AStack    ENDS

                          ; Данные программы

 0000                     DATA      SEGMENT

                          ;  Директивы описания данн
                          ых

 0000  0003               a      DW    3
 0002  0001               b      DW    1
 0004  0001               i      DW    1
 0006  FFFF               k      DW    -1
 0008  0000               i1     DW    0
 000A  0000               i2     DW    0
 000C  0000               res    DW    0

 000E                     DATA      ENDS

                          ; Код программы

 0000                     CODE      SEGMENT
                                    ASSUME CS:CODE, DS:DATA, SS:AStack

                          ; Головная процедура
```

```
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 16 0000 R                   mov dx,a
0010  3B 16 0002 R                   cmp dx,b
0014  7E 1E                          jle l1
                                     ; a>b
                                            ;первая функэ
                        ĭ○Ú
0016  D1 E0                          sal ax,1
0018  D1 E0                          sal ax,1
001A  05 0003                              add ax,3
Microsoft (R) Macro Assembler Version 5.10                  11/7/18 06:28:06
                                                             Page     1-2


001D  F7 D8                          neg ax
001F  A3 0008 R                      mov i1,ax
0022  A1 0004 R                      mov ax,i
                                     ;вторая функэ
                        ĭ○Ú
0025  D1 E0                          sal ax,1
0027  D1 E0                          sal ax,1
0029  F7 D8                          neg ax
002B  05 0007                              add ax, 7
002E  A3 000A R                      mov i2,ax
0031  EB 30 90                            jmp l1e
0034                                 l1:
                                     ; a<=b
                                            ;первая функэ
                        ĭ○Ú
0034  D1 E0                         sal ax,1
0036  D1 E0                           sal ax,1
0038  03 06 0004 R                    add ax,i
003C  03 06 0004 R                    add ax,i
0040  F7 D8                           neg ax
0042  05 000A                              add ax,10
0045  F7 D8                           neg ax
0047  A3 0008 R                      mov i1,ax
004A  A1 0004 R                      mov ax,i
                                     ;вторая функэ
                        ĭ○Ú
```

```
004D  D1 E0                          sal ax,1
004F  D1 E0                            sal ax,1
0051  03 06 0004 R                     add ax,i
0055  03 06 0004 R                     add ax,i
0059  F7 D8                            neg ax
005B  05 0008                                add ax,8
005E  F7 D8                            neg ax
0060  A3 000A R                        mov i2, ax
0063                               l1e:

0063  83 3E 0006 R 00                  cmp k,0 ; третья функ
                     ция
0068  7C 1C                        jl l2
                                   ; k>=0
006A  A1 000A R                        mov ax,i2
006D  3D 0000                                cmp ax,0
0070  79 02                            jns ns
0072  F7 D8                                  neg ax ;еслЍ
                     ِi2<0
0074                                   ns:
0074  F7 D8                            neg ax
0076  05 0003                                add ax,3
0079  F7 D8                            neg ax
007B  3D 0004                                cmp ax, 4
007E  7F 03                            jg max ; ax>4 - перЍ
                     ̂ƐƩՐ◌ٍآна метку max
0080  B8 0004                                      mov ax, 4 ; иЍ
                     ـــёȒ?кладем в ax 4
```



```
0083                                   max:
                                       ;максиммум
0083  EB 24 90                               jmp l2e
0086                               l2:
                                   ; k<0
0086  A1 000A R                        mov ax, i2
0089  83 3E 000A R 00                        cmp i2,0
008E  79 02                            jns gi2
0090  F7 D8                                  neg ax ;еслЍ
                     ِi2<0
0092                                   gi2:
0092  83 3E 0008 R 00                        cmp i1,0
0097  79 02                            jns gi1
0099  F7 D8                                  neg ax ;еслЍ
                     ِi1<0 делаем i2<0
```

```
 009B                                           gi1:
 009B  03 06 0008 R                             add ax,i1
 009F  3D 0000                                        cmp ax,0
 00A2  79 02                                    jns sum
 00A4  F7 D8                                          neg ax ;еслЙ
                        يi1+i2<0
 00A6                                           sum:
 00A6  A3 000C R                                mov res,ax
 00A9                                   l2e:

 00A9  A3 000C R                        mov res,ax ;кладем зна
                        чение третьей функции
 00AC  CB                                       ret
 00AD                   Main    ENDP
 00AD                   CODE    ENDS
                                END Main
Microsoft (R) Macro Assembler Version 5.10                  11/7/18 06:28:06
                                                             Symbols-1


Segments and Groups:

                N a m e         Length  Align Combine Class

ASTACK . . . . . .              0018   PARA   STACK
CODE . . . . . .                00AD   PARA   NONE
DATA . . . . . .                000E   PARA   NONE

Symbols:

                N a m e         Type    Value  Attr

A  . . . . . .                  L WORD 0000   DATA

B  . . . . . .                  L WORD 0002   DATA

GI1  . . . . . .                L NEAR 009B   CODE
GI2  . . . . . .                L NEAR 0092   CODE

I  . . . . . .                  L WORD 0004   DATA
I1 . . . . . .                  L WORD 0008   DATA
I2 . . . . . .                  L WORD 000A   DATA

K  . . . . . .                  L WORD 0006   DATA

L1 . . . . . .                  L NEAR 0034   CODE
L1E  . . . . . .                L NEAR 0063   CODE
```

```
L2 . . . . . .      L NEAR 0086   CODE
L2E  . . . . . .    L NEAR 00A9   CODE

MAIN . . . . . .    F PROC 0000   CODE   Length = 00AD
MAX  . . . . . .    L NEAR 0083   CODE

NS . . . . . .      L NEAR 0074   CODE

RES  . . . . . .    L WORD 000C   DATA

SUM  . . . . . .    L NEAR 00A6   CODE

@CPU . . . . . .    TEXT  0101h
@FILENAME  . . . .  TEXT  LR3
@VERSION . . . . .  TEXT  510

Microsoft (R) Macro Assembler Version 5.10                  11/7/18 06:28:06

                                                             Symbols-2

    120 Source  Lines
    120 Total   Lines
     25 Symbols

  48030 + 459230 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```