

**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. 9382

\_\_\_\_\_

Кодуков А.В.

Преподаватель

\_\_\_\_\_

Ефремов М.А .

Санкт-Петербург

2020

**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  $b$ .

Вариант 10: (2, 5, 6)

$/ - (4*i+3)$  , при  $a>b$

$f1 = f2 = <$

$\backslash 6*i - 10$  , при  $a \leq b$

$/ 20 - 4*i$  , при  $a>b$

$f2 = f5 = <$

$\backslash -(6*I - 6)$ , при  $a \leq b$

$/ |i1 - i2|$ , при  $k < 0$

$f3 = f6 = <$

$\backslash \max(7, |i2|)$ , при  $k \geq 0$

**Выполнение работы:**

Использованные для ветвления и вычислений команды:

*shl* - логический сдвиг влево

*cmp* – сравнение с изменением флагов

*jmp* – косвенный переход

*jg, jnl...* - условный переход

Тестирование:

- 1)  $a > b, k \geq 0$
- 2)  $a > b, k < 0$
- 3)  $a \geq b, k \geq 0$
- 4)  $a \geq b, k < 0$

№	A	B	i	k	f1	f2	f3
1	2	1	1	1	-7	16	16
2	9	8	2	-1	-11	12	23
3	1	1	1	1	-4	0	7
4	1	8	2	-1	2	-6	8

**Вывод:**

В ходе выполнения работы были изучены команды ветвления и работы с целыми числами, реализована программа вычисления кусочно-заданных функций.

## ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

### branch.asm

```
STACKSG SEGMENT PARA STACK 'Stack'

DW 32 DUP(?)

STACKSG ENDS

DATASG SEGMENT PARA 'Data' ;SEG DATA

VARA DW 1h
VARB DW 1h
VARI DW 1h
VARK DW 1h
VARI1 DW 1h
VARI2 DW 1h
VARRES DW 1h

DATASG ENDS ;ENDS DATA

CODE SEGMENT ;SEG CODE

ASSUME DS:DataSG, CS:Code

Main PROC FAR

mov ax, DATASG ;ds setup
mov ds, ax

f1: ;dw f1(VARA,VARB,VARI)

mov ax,VARA ;ax = *si (a)
mov si,VARB ;si = &b
mov bx,VARI ;bx = i
shl bx,1 ;bx *= 2
cmp ax,si ;cmp ax,*si (a,b)
jg f1_1 ;if a>b

f1_2: ;else (6i-10)

shl bx,1 ;bx *= 2
mov ax,VARI ;ax = i
shl ax,1 ;ax *= 2
add ax,bx ;ax += bx (ax = 6i)
sub ax,10 ;ax -= 10
push ax ;ret ax
jmp f1_end

f1_1: ;then -(4i+3) = (-4i - 3)

shl bx,1 ;bx *= 2
sub ax,ax ;ax = 0
sub ax,bx ;ax = -bx (-4i)
sub ax,3 ;ax -= 3
push ax ;ret ax

f1_end:
```

```

mov ax,VARI1
pop VARI1

f2: ;dw f2(VARA,VARB,VARI)

mov ax,VARA ;ax = *si (a)
cmp ax,VARB ;cmp ax,*si (a,b)
jg f2_1 ;if a>b
jmp f2_2

f2_1: ;then 20 - 4i

mov bx,VARI ;bx = i
shl bx,1 ;bx *= 2
shl bx,1 ;bx *= 2
mov ax,bx
sub bx,bx ;bx = 0
sub bx,ax ;bx = -ax (bx = -4i)
add bx,20 ;bx += 20
push bx ;ret bx
jmp f2_end

f2_2: ;else 6(1 - i) = -(6i-6)

sub ax,ax; ax = 0
sub ax,VARI ;ax = -i
add ax,1; ax = 1 - i
shl ax,1 ;ax *= 2
mov bx,ax; bx = 2(1-i)
shl ax,1; ax *= 2
add ax,bx ;ax += 2(1-i)
push ax ;ret ax

f2_end:

pop VARI2

f3: ;dw f3(VARI1,VARI2,VARK)

mov ax,VARK ;ax = *si (k)
cmp ax,0 ;cmp k,0
jl f3_1 ;if k < 0

f3_2: ;else max(7, abs(i2))

mov ax,VARI2 ;ax = i2
cmp ax,0 ;cmp i2,0
jnl f3_2_c ;unless ax >= 0
neg ax ;ax *= -1
f3_2_c: ;else dont
cmp ax,7
jnl f3_2_d ;unless ax >= 7
mov ax,7 ;ax = 7
f3_2_d:
push ax ;ret i2
jmp f3_end

f3_1: ;then abs(i1-i2)

mov ax,VARI1 ;ax = i1
sub ax,VARI2 ;ax -= i2 (ax = i1 - i2)
cmp ax,0 ;cmp (i1 - i2),0
jnl f3_1_c ;unless ax >= 0

```

```

neg ax ;neg ax
f3_1_c: ;else dont
push ax ;ret ax

f3_end:

pop VARRES
mov ah, 4ch ;exit
int 21h

Main ENDP
CODE ENDS
END Main

```

## ПРИЛОЖЕНИЕ В ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Microsoft (R) Macro Assembler Version 5.10

10/14/20 22:07:4

Page 1-1

```

0000                                STACKSG SEGMENT PARA STACK 'Stack'

0000 0020[                            DW 32 DUP(?)
      ????                            ]

0040                                STACKSG ENDS

0000                                DATASG SEGMENT PARA 'Data' ;SEG DATA

0000 0001                            VARA DW 1h
0002 0001                            VARB DW 1h
0004 0001                            VARI DW 1h
0006 0001                            VARK DW 1h
0008 0001                            VARI1 DW 1h
000A 0001                            VARI2 DW 1h
000C 0001                            VARRES DW 1h

000E                                DATASG ENDS ;ENDS DATA

0000                                CODE SEGMENT ;SEG CODE

                                ASSUME DS:DataSG, CS:Code

0000                                Main PROC FAR

0000 B8 ---- R                        mov ax, DATASG ;ds setup
0003 8E D8                            mov ds, ax

0005                                f1: ;dw f1(VARA,VARB,VARI)

0005 A1 0000 R                        mov ax,VARA ;ax = *si (a)
0008 8B 36 0002 R                    mov si,VARB ;si = &b
000C 8B 1E 0004 R                    mov bx,VARI ;bx = i
0010 D1 E3                            shl bx,1 ;bx *= 2
0012 3B C6                            cmp ax,si ;cmp ax,*si (a,b)
0014 7F 10                            jg f1_1 ;if a>b

0016                                f1_2: ;else (6i-10)

0016 D1 E3                            shl bx,1 ;bx *= 2

```

```

0018 A1 0004 R      mov ax,VARI ;ax = i
001B D1 E0          shl ax,1 ;ax *= 2
001D 03 C3          add ax,bx ;ax += bx (ax = 6i)
001F 2D 000A        sub ax,10 ;ax -= 10
0022 50            push ax ;ret ax
0023 EB 0B 90        jmp f1_end

```

```

0026                f1_1: ;then  $-(4i+3) = (-4i - 3)$ 

```

```

0026 D1 E3          shl bx,1 ;bx *= 2
0028 2B C0          sub ax,ax ;ax = 0
002A 2B C3          sub ax,bx ;ax = -bx (-4i)

```

Microsoft (R) Macro Assembler Version 5.10

10/14/20 22:07:4

Page 1-2

```

002C 2D 0003        sub ax,3 ;ax -= 3
002F 50            push ax ;ret ax

```

```

0030                f1_end:

```

```

0030 A1 0008 R      mov ax,VARI1
0033 8F 06 0008 R  pop VARI1

```

```

0037                f2: ;dw f2(VARA,VARB,VARI)

```

```

0037 A1 0000 R      mov ax,VARA ;ax = *si (a)
003A 3B 06 0002 R  cmp ax,VARB ;cmp ax,*si (a,b)
003E 7F 03          jg f2_1 ;if a>b
0040 EB 16 90        jmp f2_2

```

```

0043                f2_1: ;then  $20 - 4i$ 

```

```

0043 8B 1E 0004 R  mov bx,VARI ;bx = i
0047 D1 E3          shl bx,1 ;bx *= 2
0049 D1 E3          shl bx,1 ;bx *= 2
004B 8B C3          mov ax,bx
004D 2B DB          sub bx,bx ;bx = 0
004F 2B D8          sub bx,ax ;bx = -ax (bx = -4i)
0051 83 C3 14        add bx,20 ;bx += 20
0054 53            push bx ;ret bx
0055 EB 13 90        jmp f2_end

```

```

0058                f2_2: ;else  $6(1 - i) = -(6i-6)$ 

```

```

0058 2B C0          sub ax,ax; ax = 0
005A 2B 06 0004 R  sub ax,VARI ;ax = -i
005E 05 0001        add ax,1; ax = 1 - i
0061 D1 E0          shl ax,1 ;ax *= 2
0063 8B D8          mov bx,ax; bx = 2(1-i)
0065 D1 E0          shl ax,1; ax *= 2
0067 03 C3          add ax,bx ;ax += 2(1-i)
0069 50            push ax ;ret ax

```

```

006A                f2_end:

```

```

006A 8F 06 000A R  pop VARI2

```

```

006E                f3: ;dw f3(VARI1,VARI2,VARK)

```

```

006E A1 0006 R      mov ax,VARK ;ax = *si (k)
0071 3D 0000        cmp ax,0 ;cmp k,0

```

```

0074 7C 16                j1 f3_1 ;if k < 0
0076                    f3_2: ;else max(7, abs(i2))
0076 A1 000A R            mov ax,VARI2 ;ax = i2
0079 3D 0000              cmp ax,0 ;cmp i2,0
007C 7D 02                jnl f3_2_c ;unless ax >= 0
007E F7 D8                neg ax ;ax *= -1

```

Microsoft (R) Macro Assembler Version 5.10

10/14/20 22:07:4

Page 1-3

```

0080                    f3_2_c: ;else dont
0080 3D 0007              cmp ax,7
0083 7D 03                jnl f3_2_d ;unless ax >= 7
0085 B8 0007              mov ax,7 ;ax = 7
0088                    f3_2_d:
0088 50                    push ax ;ret i2
0089 EB 10 90             jmp f3_end

008C                    f3_1: ;then abs(i1-i2)

008C A1 0008 R            mov ax,VARI1 ;ax = i1
008F 2B 06 000A R        sub ax,VARI2 ;ax -= i2 (ax = i1 - i2)
0093 3D 0000              cmp ax,0 ;cmp (i1 - i2),0
0096 7D 02                jnl f3_1_c ;unless ax >= 0
0098 F7 D8                neg ax ;neg ax
009A                    f3_1_c: ;else dont
009A 50                    push ax ;ret ax

009B                    f3_end:

009B 8F 06 000C R        pop VARRES
009F B4 4C                mov ah, 4ch ;exit
00A1 CD 21                int 21h

00A3                    Main ENDP
00A3                    CODE ENDS
00A3                    END Main

```



Segments and Groups:

N a m e	Length	Align	Combine Class
CODE . . . . .	00A3	PARA	NONE
DATASG . . . . .	000E	PARA	NONE 'DATA'
STACKSG . . . . .	0040	PARA	STACK 'STACK'

Symbols:

N a m e	Type	Value	Attr
F1 . . . . .	L NEAR	0005	CODE
F1_1 . . . . .	L NEAR	0026	CODE
F1_2 . . . . .	L NEAR	0016	CODE
F1_END . . . . .	L NEAR	0030	CODE
F2 . . . . .	L NEAR	0037	CODE
F2_1 . . . . .	L NEAR	0043	CODE
F2_2 . . . . .	L NEAR	0058	CODE
F2_END . . . . .	L NEAR	006A	CODE
F3 . . . . .	L NEAR	006E	CODE
F3_1 . . . . .	L NEAR	008C	CODE
F3_1_C . . . . .	L NEAR	009A	CODE
F3_2 . . . . .	L NEAR	0076	CODE
F3_2_C . . . . .	L NEAR	0080	CODE
F3_2_D . . . . .	L NEAR	0088	CODE
F3_END . . . . .	L NEAR	009B	CODE
MAIN . . . . .	F PROC	0000	CODE Length = 00A3
VARA . . . . .	L WORD	0000	DATASG
VARB . . . . .	L WORD	0002	DATASG
VARI . . . . .	L WORD	0004	DATASG
VARI1 . . . . .	L WORD	0008	DATASG
VARI2 . . . . .	L WORD	000A	DATASG
VARK . . . . .	L WORD	0006	DATASG
VARRES . . . . .	L WORD	000C	DATASG
@CPU . . . . .	TEXT	0101h	
@FILENAME . . . . .	TEXT	branch	
@VERSION . . . . .	TEXT	510	

132 Source Lines  
132 Total Lines  
33 Symbols

48042 + 459218 Bytes symbol space free

0 Warning Errors  
0 Severe Errors