

**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

Желнин М.Ю.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### Цель работы.

Изучить механизм работы представлений и обработки целых чисел.

### 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 6

$$f1 = \begin{cases} / 15-2*i, & \text{при } a>b \\ \backslash 3*i+4, & \text{при } a\leq b \end{cases}$$

$$f7 = \begin{cases} / -(4*i-5), & \text{при } a>b \\ \backslash 10-3*i, & \text{при } a\leq b \end{cases}$$

$$f6 = \begin{cases} / |i1-i2|, & \text{при } k<0 \\ \backslash \max(7, |i2|), & \text{при } k\geq 0 \end{cases}$$

### Выполнение работы.

Происходит расчет функций  $f_1$ ,  $f_2$ ,  $f_3$ . При организации ветвящихся процессов использовалась функция `str` и условные переходы. Для операций умножения использовался побитовый сдвиг влево и сложение.

Разработанный программный код см. в приложении А.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Проверка работы программы с помощью отладчика (все результаты заносились в реестр CX)

№ п/п	Входные данные	Полученное значение i1	Полученное значение i2	Полученное значение res	Комментарии
1	a = 1 b = 2 i = 3 k = 4	000Dh = 13	0001h = 1	0007h = 7	Верно, т.к. $a \leq b$ , $3*3 + 4 = 13$ , $(10 - 3*3) = 1$ , $k \geq 0$ , $\max(7, 1) = 7$
2	a = 1 b = 2 i = 3 k = -1	000Fh = 13	0003h = 1	000Ch = 12	Верно, т.к. $a \leq b$ , $3*3 + 4 = 13$ , $(10 - 3*3) = 1$ , $k < 0$ , $ i1-i2  = 12$
3	a = 2 b = 1 i = 3 k = 0	0009h = 9	FFF9h = -7	0007h = 7	Верно, т.к. $a > b$ , $-(15 - 2*3) = 9$ , $-(4*3-5) = -7$ , $k = 1$ , $\max(7,  -7 ) = 7$
4	a = 2 b = 1 i = 3 k = -1	0009h = 9	FFF9h = -7	0010h = 16	Верно, т.к. $a > b$ , $-(15 - 2*3) = 9$ , $-(4*3-5) = -7$ , $k = -11$ , $ 9+7  = 16$

**Выводы.**

Был изучен механизм работы представлений и обработки целых чисел.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.asm

```
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
;Данные программы
DATA     SEGMENT
;Директивы описания данных
string DB    15, 15 DUP('$')
sign    DB    1
a        DW    2
b        DW    1
i        DW    3
k        DW   -1
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

;вычисление f1 и f7
    mov  cx, i
    mov  ax, cx
    shl  cx, 1
    mov  bx, b    ;
    cmp  a, bx    ; сравнение a и b
    jle  f1f7second
        neg  cx
        add  cx, 15
        mov  i1, cx
        shl  cx, 1
        add  cx, -25
        jmp  f1f7final
f1f7second:
    add  cx, ax
    add  cx, 4
    mov  i1, cx
    neg  cx
    add  cx, 14
f1f7final:
    mov  i2, cx
```

Название файла: lab3Lst.lst

11/24/21 18:55:0

```

0000                                ; PŸC,PμPε PŸCŦPsPŸCŦP°PjPjC<
0000                                AStack SEGMENT STACK
0000                                DW 12 DUP(?)
                                ]

0018                                AStack ENDS
                                ;P”P°PSPSC<Pμ PŸCŦPsPŸCŦP°PjPjC<
0000                                DATA SEGMENT
                                ;P”PεCŦPμPεC,PεPIC< PsPŸPεCŸP°PSPεCŸ PŸP°PSPSC<
                                C...
0000 0F                                string DB 15, 15 DUP('$')
000F[
24                                ]

0010 01                                sign DB 1
0011 0002                            a DW 2
0013 0001                            b DW 1
0015 0003                            i DW 3
0017 FFFF                            k DW -1
0019 0000                            i1 DW 0

```

```

001B 0000          i2    DW    0

001D              DATA    ENDS

; PъPsPr' PĭCTъPsPĭCTъP°PjPjC<
0000              CODE     SEGMENT
                        ASSUME CS:CODE, DS:DATA, SS:Astack

; P“PsP»PsPIPSP°Cĭ PĭCTъPsC†PμPrCfCTъP°
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

;PIC<C†PěCfP»PμPSPěPμ f1 Pě f7
000C 8B 0E 0015 R    mov     cx, i
0010 8B C1          mov     ax, cx
0012 D1 E1          shl     cx, 1
0014 8B 1E 0013 R    mov     bx, b ;
0018 39 1E 0011 R    cmp     a, bx ; CfCTъP°PIPSPμPSPěPμ a P
                        ě b
001C 7E 11          jle     f1f7second
001E F7 D9          neg     cx
0020 83 C1 0F                add     cx, 15
0023 89 0E 0019 R    mov     i1, cx
Microsoft (R) Macro Assembler Version 5.10
11/24/21 18:55:0
Page 1-2

0027 D1 E1          shl     cx, 1
0029 83 C1 E7          add     cx, -25
002C EB 0F 90          jmp     f1f7final
002F              f1f7second:
002F 03 C8          add     cx, ax
0031 83 C1 04          add     cx, 4
0034 89 0E 0019 R    mov     i1, cx
0038 F7 D9          neg     cx
003A 83 C1 0E          add     cx, 14
003D              f1f7final:
003D 89 0E 001B R    mov     i2, cx

;CTъP°CfCfC†PμC, f6
0041 8B 1E 0017 R    mov     bx, k
0045 83 FB 00          cmp     bx, 0
0048 7C 23          jl      f6Second

004A 8B 0E 001B R    mov     cx, i2
004E 83 F9 00          cmp     cx, 0
0051 7D 06          jge     skip2 ;PjPsPrCfP»CHъ i2
0053 F7 D9          neg     cx
0055 89 0E 001B R    mov     i2, cx
0059              skip2:

0059 8B 1E 001B R    mov     bx, i2
005D 83 FB 07          cmp     bx, 7
0060 7C 05          jl      max1

```

```

0062 8B CB                mov cx, bx          ; |i2| >= 7
0064 EB 18 90            jmp MainFinal
0067                    max1:
0067 B9 0007              mov cx, 7          ; |i2| < 7
006A EB 12 90            jmp MainFinal
006D                    f6Second:
006D 8B 0E 001B R        mov cx, i2
0071 F7 D9              neg cx
0073 03 0E 0019 R        add cx, i1
0077 83 F9 00           cmp cx, 0
007A 7D 02              jge skip1
007C F7 D9              neg cx
007E                    skip1:

007E                    MainFinal:                ; Pl cx P»PμP¶Pë
C, P·PSP°C‡PμPSPëPμ C,,CfPSPeC‡PëPë f7
007E CB                ret
007F                    Main      ENDP
007F                    CODE     ENDS
                                END Main
Microsoft (R) Macro Assembler Version 5.10
11/24/21 18:55:0
Symbols-1

```

#### Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK . . . . .	0018	PARA	STACK	
CODE . . . . .	007F	PARA	NONE	
DATA . . . . .	001D	PARA	NONE	

#### Symbols:

N a m e	Type	Value	Attr	
A . . . . .	L WORD	0011	DATA	
B . . . . .	L WORD	0013	DATA	
F1F7FINAL . . . . .	L NEAR	003D	CODE	
F1F7SECOND . . . . .	L NEAR	002F	CODE	
F6SECOND . . . . .	L NEAR	006D	CODE	
I . . . . .	L WORD	0015	DATA	
I1 . . . . .	L WORD	0019	DATA	
I2 . . . . .	L WORD	001B	DATA	
K . . . . .	L WORD	0017	DATA	
MAIN . . . . .	F PROC	0000	CODE	Length = 007F
MAINFINAL . . . . .	L NEAR	007E	CODE	
MAX1 . . . . .	L NEAR	0067	CODE	
SIGN . . . . .	L BYTE	0010	DATA	
SKIP1 . . . . .	L NEAR	007E	CODE	
SKIP2 . . . . .	L NEAR	0059	CODE	
STRING . . . . .	L BYTE	0000	DATA	
@CPU . . . . .	TEXT	0101h		
@FILENAME . . . . .	TEXT	hello3		
@VERSION . . . . .	TEXT	510		



92 Source Lines  
92 Total Lines  
24 Symbols

47962 + 461345 Bytes symbol space free

0 Warning Errors  
0 Severe Errors