

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

Сергевнин Д.В.

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 14 (3.4.2)

$$f3 = \begin{cases} / 7 - 4 * i, & \text{при } a > b \\ \backslash 8 - 6 * i, & \text{при } a \leq b \end{cases}$$

$$f4 = \begin{cases} / -(6 * i - 4), & \text{при } a > b \\ \backslash 3 * (i + 2), & \text{при } a \leq b \end{cases}$$

$$f2 = \begin{cases} / \max(i1, 10 - i2), & \text{при } k < 0 \\ \backslash |i1 - i2|, & \text{при } k \geq 0 \end{cases}$$

Выполнение работы.

Происходит расчет функций f_3 , f_4 , f_2 . 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 i_1	Полученное значение i_2	Полученное значение res	Комментарии
1	$a = 1$ $b = 2$ $i = 3$ $k = 4$	$000Fh = 15$	$FFF6h = -10$	$0019h = 25$	Верно, т.к. $a \leq b$, $8 - 6*3 = -10$, $3*(3+2) = 15$, $k \geq 0$, $ 15 - (-10) = 25$
2	$a = 1$ $b = 2$ $i = 3$ $k = -1$	$000Fh = 15$	$FFF6h = -10$	$FFFBh = -5$	Верно, т.к. $a \leq b$, $8 - 6*3 = -10$, $3*(3+2) = 15$, $k < 0$, $\max(-10, -5) = -5$
3	$a = 2$ $b = 1$ $i = 3$ $k = 0$	$FFFBh = -5$	$FFF2h = -14$	$0009h = 9$	Верно, т.к. $a > b$, $7 - 4*3 = -5$, $-(6*3 - 4) = -14$, $k \geq 0$, $ -5 - (-14) = 9$
4	$a = 2$ $b = 1$ $i = 3$ $k = -1$	$FFFBh = -5$	$FFF2h = -14$	$0018h = 24$	Верно, т.к. $a > b$, $7 - 4*3 = -5$, $-(6*3 - 4) = -14$, $k < 0$, $\max(-5, 24) = 24$

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
a      DW      1
b      DW      3
i      DW      2
k      DW      3
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

;3.4.2
;вычисление f3 и f4
    mov    cx, i
    mov    ax, cx
    add    ax, cx ; 2i
    mov    bx, b
    cmp    a, bx ;сравнение a и b
    jg     f34 ;a<=b
    add    cx, ax ;3i
    add    cx, 6
    mov    i2, cx ;3i + 6
    neg    cx
    shl    cx, 1 ; -6i - 12
    add    cx, 20
    mov    i1, cx ;8 -6*i
    jmp    final
```

```

f34: ;при a>b
    mov cx, ax
    neg cx      ; -2i
    shl cx, 1   ; -4i
    add cx, 7
    mov i1, cx ; 7 - 4*i
    shl cx, 1   ; -8i+14
    add cx, ax ; -6i+14
    add cx, -10
    mov i2, cx ; -6*i + 4

final: ;вычисление f2
    mov ax, i1
    mov cx, i2
    neg cx
    mov bx, k
    cmp bx, 0
    jge f2
    add cx, 10 ; max(i1, 10-i2), при k<0
    cmp cx, ax
    jg step1
    mov cx, ax ; если ax было > cx
    jmp MainFinal
step1:
    jmp MainFinal

f2: ; |i1 - i2| , при k>=0
    add cx, ax
    cmp cx, 0
    jg step2
    neg cx
step2:
MainFinal: ; в cx лежит значение функции f2
    ret
Main      ENDP
CODE      ENDS
END Main

```

Название файла: lab3.lst

Microsoft	(R)	Macro	Assembler	Version	5.10
11/22/21 18:08:4					
					Page
1-1					

```

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

0018      AStack  ENDS

```

```

;Данные программы
0000          DATA          SEGMENT
;Директивы описания данны
x
0000 0001          a          DW          1
0002 0003          b          DW          3
0004 0002          i          DW          2
0006 0003          k          DW          3
0008 0000          i1         DW          0
000A 0000          i2         DW          0

000C          DATA          ENDS

; Код программы
0000          CODE          SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

; Главная 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 B9 0000          mov CX, 0

;3.4.2
;вычисление f3 и f4
000C 8B 0E 0004 R          mov cx, i
0010 8B C1          mov ax, cx
0012 03 C1          add ax, cx ; 2i
0014 8B 1E 0002 R          mov bx, b
0018 39 1E 0000 R          cmp a, bx ;сравнение a и b
001C 7F 17          jg f34 ;a<=b
001E 03 C8          add cx, ax ;3i
0020 83 C1 06          add cx, 6
0023 89 0E 000A R          mov i2, cx ;3i + 6
0027 F7 D9          neg cx
0029 D1 E1          shl cx, 1 ; -6i - 12
002B 83 C1 14          add cx, 20
002E 89 0E 0008 R          mov i1, cx ;8 -6*i
0032 EB 19 90          jmp final

```

```

0035          f34: ;при a>b

```

```

0035 8B C8          mov cx, ax
0037 F7 D9          neg cx      ; -2i
0039 D1 E1          shl cx, 1   ; -4i
003B 83 C1 07        add cx, 7
003E 89 0E 0008 R    mov i1, cx ; 7 - 4*i
0042 D1 E1          shl cx, 1   ; -8i+14
0044 03 C8          add cx, ax  ; -6i+14
0046 83 C1 F6        add cx, -10
0049 89 0E 000A R    mov i2, cx ; -6*i + 4

004D              final: ;вычисление f2
004D A1 0008 R        mov ax, i1
0050 8B 0E 000A R    mov cx, i2
0054 F7 D9          neg cx
0056 8B 1E 0006 R    mov bx, k
005A 83 FB 00        cmp bx, 0
005D 7D 0F          jge f2
005F 83 C1 0A        add cx, 10 ;max(i1,10-i2),
                    при k<0
0062 3B C8          cmp cx, ax
0064 7F 05          jg step1
0066 8B C8          mov cx, ax   ;если ax
                    было > cx
0068 EB 0D 90        jmp MainFinal
006B              step1:
006B EB 0A 90        jmp MainFinal

006E              f2: ; |i1 - i2| , при k>=0
006E 03 C8          add cx, ax
0070 83 F9 00        cmp cx, 0
0073 7F 02          jg step2
0075 F7 D9          neg cx
0077              step2:
0077              MainFinal:      ; в cx лежи
                    т значение функции f2
0077 CB            ret
0078              Main          ENDP
0078              CODE          ENDS
                    END Main

```

```

Microsoft (R) Macro Assembler Version 5.10
11/22/21 18:08:4

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	0018	PARA	STACK
CODE	0078	PARA	NONE
DATA	000C	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F2	L NEAR	006E	CODE
F34	L NEAR	0035	CODE
FINAL	L NEAR	004D	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
= 0078	MAINFINAL	L NEAR	0077	CODE
STEP1	L NEAR	006B	CODE
STEP2	L NEAR	0077	CODE
@CPU	TEXT	0101h	
@FILENAME	TEXT	lab3	
@VERSION	TEXT	510	

86 Source Lines
86 Total Lines
21 Symbols

47978 + 461329 Bytes symbol space free

0 Warning Errors
0 Severe Errors