

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
КАФЕДРА МО ЭВМ

ОТЧЕТ

по лабораторной работе №3
по дисциплине «ОргЭВМиС»

Тема: Представление и обработка целых чисел. Организация
ветвящихся процессов.

Студентка гр. 0382

Морева Е.С.

Преподаватели

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить представление и обработку целых чисел. Научиться организовывать ветвящиеся процессы на языке Ассемблера.

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 $res = f3(i1, i2, k)$,

$f1 = -(4i + 3)$, при $a > b$;

$6i - 10$, при $a \leq b$

$f2 = -(6i - 4)$, при $a > b$;

$3 \cdot (i + 2)$, при $a \leq b$

$f3 = |i1| + |i2|$, при $k < 0$;

$\max(6, |i1|)$, при $k \geq 0$

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 .

Выполнение работы.

В регистре cl хранится значение a для последующего сравнения с b , а в al и bl — значение i , для вычисления функций $f1$ и $f2$. Так как функции $f1$ $f2$ имеют одинаковые условия на a и b , вычисляем их в одно время. В функции $f3$ используются абсолютные значения функций $f1$ $f2$, поэтому регистры al bl , хранящие эти значения, возьмем по модулю. В зависимости от k вычислим значение $f3$. Завершим программу.

Разработанный программный код см. в приложении А.

Файл листинга см. в приложении В.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	i1	i2	Результат	Комментарии
1.	a = -3 b = 7 i = -2 k = 1	-24	0	f3 = 6	Верно. $a < b \Rightarrow$ $i1 = 6 * (-2) - 10 = -24;$ $i2 = 3 * (-2 + 2) = 0;$ $k \geq 0 \Rightarrow \max(6, i1) = 6$
2.	a = -3 b = 7 i = -2 k = -1	-24	0	f3 = 24	Верно. $a < b \Rightarrow$ $i1 = 6 * (-2) - 10 = -24;$ $i2 = 3 * (-2 + 2) = 0;$ $k < 0 \Rightarrow i1 + i2 = 24$
3.	a = 0 b = -1 i = 2 k = -3	-11	8	f3 = 19	Верно. $a > b \Rightarrow$ $i1 = -(4 * 2 + 3) = -11;$ $i2 = -(6 * 2 - 4) = 8;$ $k < 0 \Rightarrow i1 + i2 = 19$

Выводы.

Были изучены представление и обработка целых чисел. Получены знания об организации ветвящихся процессов на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЕ КОДЫ ПРОГРАММ

Название файла: lr3.asm

```
.MODEL MEDIUM

.STACK 200h

.data
a db 4
b db 3
k db 2
i db 1

.CODE
mov ax,@data
mov ds, ax
xor ax, ax

mov cl, a
mov al, i ; al=i
mov bl, i ; al=i

cmp cl, b
JG L1
    sal al, 1 ; al=2*i
    add al, i ; al=3*i
    sal al, 1 ; al=6*i
    sub al, 10 ; al=6*i-10

    add bl, 2 ; bl=i+2
    mov cl, bl
    add bl, bl ; bl=2*(i+2)
    add bl, cl ; bl=3*(i+2)
jmp endL1

L1:
    sal al, 1 ; al=2*i
    sal al, 1 ; al=4*i
    add al, 3 ; al=4*i+3
    neg al ; al=-(4*i+3)

    sal bl, 1 ; bl=2*i
    add bl, i ; bl=3*i
    sal bl, 1 ; bl=6*i
    sub bl, 4 ; bl=6*i-4
    neg bl ; bl=-(6*i-4)
endL1:

getabsI1:
    neg al
js getabsI1 ; al=|i1|

getabsI2:
    neg bl
```

```
js getabsI2          ; bl=|i2|
```

```
cmp k, 0
jge L2
    add al,b1
    jmp endL2
L2:
    cmp al, 6
    jg max
    mov al, 6
max:
endL2:

    mov ah, 4ch
    int 21h
ret
end
```

ПРИЛОЖЕНИЕ В

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr3.lst

Microsoft (R) Macro Assembler Version 5.10

10/20/21 25:23:1

Page 1-1

```

.MODEL MEDIUM

.STACK 200h

.data
0000 04      a db 4
0001 03      b db 3
0002 02      k db 2
0003 01      i db 1

.CODE

0000 B8 ---- R      mov ax,@data
0003 8E D8      mov ds, ax
0005 33 C0      xor ax, ax

0007 8A 0E 0000 R      mov cl, a
000B A0 0003 R      mov al, i ; al=i
000E 8A 1E 0003 R      mov bl, i ; al=i

0012 3A 0E 0001 R      cmp cl, b
0016 7F 16      JG L1
0018 D0 E0      sal al, 1 ; al=2*i
001A 02 06 0003 R      add al, i ; al=3*i
001E D0 E0      sal al, 1 ; al=6*i
0020 2C 0A      sub al, 10 ; al=6*i-10

0022 80 C3 02      add bl, 2 ; bl=i+2
0025 8A CB      mov cl, bl
0027 02 DB      add bl, bl ; bl=2*(i+2)
0029 02 D9      add bl, cl ; bl=3*(i+2)
002B EB 16 90      jmp endL1

002E          L1:
002E D0 E0      sal al, 1 ; al=2*i
0030 D0 E0      sal al, 1 ; al=4*i
0032 04 03      add al, 3 ; al=4*i+3
0034 F6 D8      neg al ; al=-(4*i+3)

0036 D0 E3      sal bl, 1 ; bl=2*i
0038 02 1E 0003 R      add bl, i ; bl=3*i
003C D0 E3      sal bl, 1 ; bl=6*i
003E 80 EB 04      sub bl, 4 ; bl=6*i-4
0041 F6 DB      neg bl ; bl=-(6*i-4)
0043          endL1:

0043          getabsI1:
0043 F6 D8      neg al
0045 78 FC      js getabsI1 ; al=|i1|

0047          getabsI2:
0047 F6 DB      neg bl
0049 78 FC      js getabsI2 ; bl=|i2|

```

```

004B 80 3E 0002 R 00      cmp k, 0
0050 7D 05                jge L2
0052 02 C3                add al,b1
0054 EB 07 90              jmp endL2
0057                      L2:
0057 3C 06                cmp al, 6
0059 7F 02                jg max
005B B0 06                mov al, 6
005D                      max:
005D                      endL2:

005D B4 4C                mov ah, 4ch
005F CD 21                int 21h
0061 C3                  ret
                        end

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0004	WORD	PUBLIC	'DATA'
STACK	0200	PARA	STACK	'STACK'
LAB3_TEXT	0062	WORD	PUBLIC	'CODE'

Symbols:

N a m e	Type	Value	Attr
A	L BYTE	0000	_DATA
B	L BYTE	0001	_DATA
ENDL1	L NEAR	0043	LAB3_TEXT
ENDL2	L NEAR	005D	LAB3_TEXT
GETABSI1	L NEAR	0043	LAB3_TEXT
GETABSI2	L NEAR	0047	LAB3_TEXT
I	L BYTE	0003	_DATA
K	L BYTE	0002	_DATA
L1	L NEAR	002E	LAB3_TEXT
L2	L NEAR	0057	LAB3_TEXT
MAX	L NEAR	005D	LAB3_TEXT
@CODE	TEXT	lab3_TEXT	
@CODESIZE	TEXT	1	
@CPU	TEXT	0101h	
@DATASIZE	TEXT	0	
@FILENAME	TEXT	lab3	
@VERSION	TEXT	510	

69 Source Lines
69 Total Lines
28 Symbols

48042 + 461265 Bytes symbol space free

0 Warning Errors
0 Severe Errors


```
0050 72 0A                jb L2
0052 80 FB 06            cmp bl, 6
0055 7F 02                jg max
0057 B3 06                mov bl, 6
0059                    max:
0059 EB 03 90            jmp endL2
005C                    L2:
005C 02 D8                add bl,al
005E                    endL2:

005E 8A D3                mov dl, bl
0060 B4 02                mov ah, 02
0062 CD 21                int 21h

0064 B4 4C                mov ah, 4ch
0066 CD 21                int 21h
0068 C3                ret
                        end
```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0004	WORD	PUBLIC	'DATA'
STACK	0200	PARA	STACK	'STACK'
LAB3_TEXT	0069	WORD	PUBLIC	'CODE'

Symbols:

N a m e	Type	Value	Attr
A	L BYTE	0000	_DATA
B	L BYTE	0001	_DATA
ENDL1	L NEAR	0043	LAB3_TEXT
ENDL2	L NEAR	005E	LAB3_TEXT
GETABSI1	L NEAR	0043	LAB3_TEXT
GETABSI2	L NEAR	0047	LAB3_TEXT
I	L BYTE	0003	_DATA
K	L BYTE	0002	_DATA

L1 L NEAR 002E LAB3_TEXT

L2 L NEAR 005C LAB3_TEXT

MAX L NEAR 0059 LAB3_TEXT

@CODE TEXT lab3_TEXT

@CODESIZE TEXT 1

@CPU TEXT 0101h

@DATASIZE TEXT 0

@FILENAME TEXT lab3

@VERSION TEXT 510

72 Source Lines

72 Total Lines

28 Symbols

48046 + 461261 Bytes symbol space free

0 Warning Errors

0 Severe Errors