

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

Студентка гр. 0382

Охотникова Г.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Целью данной работы является изучение представления и обработки целых чисел, организации ветвящихся процессов.

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.

Выполнение работы.

Вариант 12:

$$f2 = \begin{cases} /-(4*i+3), & \text{при } a > b \\ \backslash 6*i-10, & \text{при } a \leq b \end{cases} \quad f7 = \begin{cases} /-(4*i-5), & \text{при } a > b \\ \backslash 10-3*i, & \text{при } a \leq b \end{cases} \quad f4 = \begin{cases} / \min(|i1-i2|, 2), & \text{при } k < 0 \\ \backslash \max(-6, -i2), & \text{при } k \geq 0 \end{cases}$$

При выполнении данной лабораторной работы в программе было создано три сегмента: сегмент стека — AStack, сегмент данных — DATA SEGMENT и сегмент кода — CODE SEGMENT. В сегменте данных были объявлены переменные a , b , I , k , $i1$, $i2$ и res .

В сегменте кода реализована процедура Main, в которой происходит вычисление значений функций. Команда `jle` позволяет совершить условный переход во вторую «опцию» функций при сравнении a и b в том случае, если a меньше или равно b . С помощью команды `jmp` осуществляются безусловные переходы для записи результатов вычислений из регистровых сегментов в переменные.

При вычислении результирующей функции используется условный переход `jge`, если левый операнд больше или равен правого. При сравнении k с нулем определяется, что будет записано в результат res .

Исходный программный код смотреть в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	a = 0 b = 2 i = 2 k = 1	i1 = 2 i2 = -4 res = -4	Программа работает корректно
2.	a = 1 b = -2 i = 5 k = -1	i1 = -23 i2 = -15 res = 2	Программа работает корректно
3.	a = 2 b = -2 i = 0 k = 5	i1 = -3 i2 = 5 res = 2	Программа работает корректно

Выводы.

При выполнении данной лабораторной работы были изучены принципы представления и обработки целых чисел, организации ветвящихся процессов.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.asm

```
AStack SEGMENT STACK
```

```
    DW 12 DUP(?)
```

```
AStack ENDS
```

```
DATA SEGMENT
```

```
    a DW 0
```

```
    b DW 2
```

```
    i DW 2
```

```
    k DW 1
```

```
    i1 DW 0
```

```
    i2 DW 0
```

```
    res DW 0
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
    ASSUME CS:CODE, DS:DATA, SS:AStack
```

```
Main PROC FAR
```

```
    push DS
```

```
    sub AX,AX
```

```
    push AX
```

```
    mov AX,DATA
```

```
    mov DS,AX
```

```
func1:
```

```
    mov ax, i
```

```
    shl ax, 1 ;2i
```

```
    shl ax, 1 ;4i
```

```
    mov bx, a
```

```
    cmp bx, b ;comparison a & b
```

```
    jle func1_2 ;jump to func1_2 if a <= b
```

```
    mov bx, ax
```

```
    add ax, 3 ;4i+3
```

```
    neg ax ;-(4i+3)
```

```

    sub bx, 5 ;4i-5
    neg bx ;-(4i-5)
    jmp func1_end

func1_2:
    shr ax, 1 ;2i
    mov bx, ax
    shl ax, 1 ;4i
    add ax, bx ;6i
    sub ax, 10 ;6i-10
    neg bx ;-2i
    sub bx, i ;-3i
    add bx, 10 ;10-3i
    jmp func1_end

func1_end:
    mov i1, ax
    mov i2, bx

func3:
    mov ax, i1
    sub ax, i2 ;i1-i2
    cmp ax, 0
    jge comp_k ;jump if (i1-i2) >= 0
    neg ax

comp_k:
    cmp k, 0
    jge func3_2

    cmp ax, 2
    jge min ;if |a1-i2|>=2
    mov bx, ax
    jmp func3_res

min:
    mov bx, 2
    jmp func3_res

```

```

func3_2:
    neg i2
    cmp i2, -6
    mov bx, i2
    jge func3_res
    mov bx, -6

func3_res:
    mov res, bx
    ret

```

```

Main ENDP
CODE ENDS
END Main

```

Название файла: lab3.lst

Microsoft (R) Macro Assembler Version 5.10

11/2/21 20:42:40

Page

1-1

```

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

```

```

0018          AStack ENDS

```

```

0000          DATA SEGMENT
0000  0000          a DW 0
0002  0002          b DW 2
0004  0002          i DW 2
0006  0001          k DW 1
0008  0000          i1 DW 0
000A  0000          i2 DW 0
000C  0000          res DW 0
000E          DATA ENDS

```

```

0000                                CODE SEGMENT

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

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                                func1:
0009 A1 0004 R                        mov ax, i
000C D1 E0                            shl ax, 1 ;2i
000E D1 E0                            shl ax, 1 ;4i
0010 8B 1E 0000 R                    mov bx, a
0014 3B 1E 0002 R                    cmp bx, b ;comparison a & b
0018 7E 0F                            jle func1_2 ;jump to func1_2 if a <=
b
                                b
001A 8B D8                            mov bx, ax
001C 05 0003                        add ax, 3 ;4i+3
001F F7 D8                            neg ax ;-(4i+3)
0021 83 EB 05                        sub bx, 5 ;4i-5
0024 F7 DB                            neg bx ;-(4i-5)
0026 EB 18 90                        jmp func1_end

0029                                func1_2:
0029 D1 E8                            shr ax, 1 ;2i
002B 8B D8                            mov bx, ax
002D D1 E0                            shl ax, 1 ;4i
002F 03 C3                            add ax, bx ;6i
0031 2D 000A                        sub ax, 10 ;6i-10
0034 F7 DB                            neg bx ; -2i
0036 2B 1E 0004 R                    sub bx, i ; -3i
003A 83 C3 0A                        add bx, 10 ;10-3i
003D EB 01 90                        jmp func1_end

0040                                func1_end:

```

20:42:40

Page

1-2

```

0040 A3 0008 R          mov i1, ax
0043 89 1E 000A R          mov i2, bx

0047                      func3:
0047 A1 0008 R          mov ax, i1
004A 2B 06 000A R          sub ax, i2 ;i1-i2
004E 3D 0000          cmp ax, 0
0051 7D 02          jge comp_k ;jump if (i1-i2) >= 0
0053 F7 D8          neg ax

0055                      comp_k:
0055 83 3E 0006 R 00          cmp k, 0
005A 7D 10          jge func3_2

005C 3D 0002          cmp ax, 2
005F 7D 05          jge min ;if |a1-i2|>=2
0061 8B D8          mov bx, ax
0063 EB 19 90          jmp func3_res

0066                      min:
0066 BB 0002          mov bx, 2
0069 EB 13 90          jmp func3_res

006C                      func3_2:
006C F7 1E 000A R          neg i2
0070 83 3E 000A R FA          cmp i2, -6
0075 8B 1E 000A R          mov bx, i2
0079 7D 03          jge func3_res
007B BB FFFA          mov bx, -6

007E                      func3_res:
007E 89 1E 000C R          mov res, bx
0082 CB          ret

```



```

0083          Main ENDP
0083          CODE ENDS
          END Main

```

Microsoft (R) Macro Assembler Version 5.10

11/2/21 20:42:40

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0083	PARA	NONE
DATA	000E	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
COMP_K	L NEAR	0055	CODE
FUNC1	L NEAR	0009	CODE
FUNC1_2	L NEAR	0029	CODE
FUNC1_END	L NEAR	0040	CODE
FUNC3	L NEAR	0047	CODE
FUNC3_2	L NEAR	006C	CODE
FUNC3_RES	L NEAR	007E	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 =
0083
MIN  . . . . . L NEAR      0066 CODE

RES  . . . . . L WORD      000C DATA

@CPU . . . . . TEXT  0101h
@FILENAME . . . . . TEXT  lab3
@VERSION . . . . . TEXT  510

```

```

88 Source  Lines
88 Total   Lines
24 Symbols

```

47978 + 461329 Bytes symbol space free

```

0 Warning Errors
0 Severe  Errors

```