

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Представление и обработка целых чисел. Организация**  
**ветвящихся процессов**

Студент гр.1381

\_\_\_\_\_

Сагидуллин Э.Р

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### Цель работы.

Изучить представление и обработку целых чисел на языке Ассемблер.

Научиться организовывать ветвящиеся процессы.

### 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 № 20

|   |   |   |
|---|---|---|
| $f4 = \begin{cases} / - (6 * i - 4) & a > b \\ \backslash 3 * (i + 2) & a \leq b \end{cases}$ | $f6 = \begin{cases} / 2 * (i + 1) - 4 & a > b \\ \backslash 5 - 3 * (i + 1) & a \leq b \end{cases}$ | $f8 = \begin{cases} /  i1  -  i2  & k < 0 \\ \backslash \max(4,  i2  - 3) & k \geq 0 \end{cases}$ |
|---|---|---|

### Выполнение работы.

Созданы три сегмента AStack, DATA, CODE - сегмент стека, сегмент кода и сегмент данных соответственно. В сегменте данных объявлены переменные  $a, b, i, k, i1, i2, res$ . В сегменте кода находится процедура Main, в которой вычисляются значения данных в условии функций. Сначала в регистрах записывается значение  $3i$ . Далее с помощью функции cmp происходит сравнение значений переменных  $a$

и b. Команда jle проверяет условие  $a \leq b$ , при его выполнении производится переход по указанному адресу. При помощи команды jne (выполнение условия  $k! = 0$ ), производится переход по указанному адресу. Для совершения безусловных переходов в программе используется команда jmp. В переменную res записывается значение регистра ax, который содержит значение функции f3.

### Тестирование.

Результаты работы программы при  $a=-3$ ;  $b=-2$ ;  $i=22$ ;  $k=-2$  представлены в табл.1.

| i1        | i2        | res      | Правильность результата |
|-----------|-----------|----------|-------------------------|
| 0048 (72) | FFC0(-64) | 0008 (8) | Верно                   |

Таблица 1 – Результаты первого теста

Результаты работы программы при  $a=2$ ;  $b=1$ ;  $i=5$ ;  $k=2$  представлены в табл.2.

| i1         | i2      | res      | Правильность результата |
|------------|---------|----------|-------------------------|
| FFE6 (-26) | 0008(8) | 0005 (5) | Верно                   |

Таблица 2 – Результаты второго теста

Результаты работы программы при  $a=4$ ;  $b=3$ ;  $i=2$ ;  $k=2$  представлены в табл.3.

| i1        | i2      | res      | Правильность результата |
|-----------|---------|----------|-------------------------|
| FFF8 (-8) | 0002(2) | 0004 (4) | Верно                   |

Таблица 3 – Результаты третьего теста

### Выводы.

В ходе выполнения лабораторной работы было изучено представление и обработка целых чисел, и организация ветвящихся процессов. Для выполнения задания была написана программа, которая вычисляет значения функций согласно заданным условиям.

## ПРИЛОЖЕНИЕ А

# ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *lab3.asm*

```
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

DATA SEGMENT
    a DW -3
    b DW -2
    i DW 22
    k DW -2
    i1 DW ?
    i2 DW ?
    res DW ?
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
f12:
    mov ax,i
    shl ax,1
    add ax,i;a=3*i
    mov cx,a
    cmp cx,b
    jle f12step2;a<=b
f12step1:
    ;f1=2*(2-3*i)
    mov i1,2
    sub i1,ax
    shl i1,1
    mov dx,i1
    ;f2=2*i-2
    sub ax,i
```

```

        mov i2,ax
        sub i2,2
        mov dx,i2
        jmp f3
f12step2:
        ;f1=3*i+6
        mov i1,ax
        add i1,6
        mov dx,i1
        ;f2=2-3*i
        mov i2,2
        sub i2,ax
        mov dx,i2
f3:
        cmp i1,0
        jge f3step1 ;i1>=0
        neg i1
f3step1:
        cmp i2,0
        jge f3step2 ;i2>=0
        neg i2
f3step2:
        cmp k,0
        jge f3step3 ; k>=0
        mov ax,i1
        sub ax,i2
        jmp f3end
f3step3:
        sub i2,3
        cmp i2,4
        jge max4 ;i2>=4
        mov ax,4
        jmp f3end
max4:
        mov ax,i2
f3end:
        mov res,ax
        ret

```

```

Main    ENDP
CODE    ENDS

END Main

```

Название файла: *lab3.lst*

Microsoft (R) Macro Assembler Version 5.10

10/30/22 22:39:1

Page 1-1

```

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

0018                                AStack ENDS

0000                                DATA SEGMENT
0000 FFFD                            a DW -3
0002 FFFE                            b DW -2
0004 0016                            i DW 22
0006 FFFE                            k DW -2
0008 0000                            i1 DW ?
000A 0000                            i2 DW ?
000C 0000                            res DW ?

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                                f12:
0009 A1 0004 R                        mov ax,i
000C D1 E0                            shl ax,1
000E 03 06 0004 R                    add ax,i;a=3*i
0012 8B 0E 0000 R                    mov cx,a

```

```

0016  3B 0E 0002 R          cmp cx,b
001A  7E 25                  jle f12step2;a<=b
001C                                f12step1:
                                ;f1=2*(2-3*i)

001C  C7 06 0008 R 0002      mov i1,2
0022  29 06 0008 R          sub i1,ax
0026  D1 26 0008 R          shl i1,1
002A  8B 16 0008 R          mov dx,i1
                                ;f2=2*i-2

002E  2B 06 0004 R          sub ax,i
0032  A3 000A R              mov i2,ax
0035  83 2E 000A R 02        sub i2,2
003A  8B 16 000A R          mov dx,i2
003E  EB 1B 90              jmp f3
0041                                f12step2:
                                ;f1=3*i+6

0041  A3 0008 R              mov i1,ax
0044  83 06 0008 R 06        add i1,6
0049  8B 16 0008 R          mov dx,i1
                                ;f2=2-3*i

004D  C7 06 000A R 0002      mov i2,2
0053  29 06 000A R          sub i2,ax
0057  8B 16 000A R          mov dx,i2
005B                                f3:

005B  83 3E 0008 R 00        cmp i1,0
0060  7D 04                  jge f3step1 ;i1>=0

```

Microsoft (R) Macro Assembler Version 5.10

10/30/22 22:39:1

Page 1-2

```

0062  F7 1E 0008 R          neg i1
0066                                f3step1:

0066  83 3E 000A R 00        cmp i2,0
006B  7D 04                  jge f3step2 ;i2>=0
006D  F7 1E 000A R          neg i2
0071                                f3step2:

0071  83 3E 0006 R 00        cmp k,0
0076  7D 0A                  jge f3step3 ; k>=0

```

```

0078  A1 0008 R          mov ax,i1
007B  2B 06 000A R      sub ax,i2
007F  EB 16 90          jmp f3end
0082                      f3step3:
0082  83 2E 000A R 03    sub i2,3
0087  83 3E 000A R 04    cmp i2,4
008C  7D 06             jge max4 ;i2>=4
008E  B8 0004           mov ax,4
0091  EB 04 90          jmp f3end
0094                      max4:
0094  A1 000A R          mov ax,i2
0097                      f3end:
0097  A3 000C R          mov res,ax
009A  CB              ret
009B                      Main  ENDP
009B                      CODE   ENDS

                      END Main

```

Microsoft (R) Macro Assembler Version 5.10

10/30/22 22:39:1

Symbols-1

Segments and Groups:

| N a m e          | Length | Align | Combine | Class |
|------------------|--------|-------|---------|-------|
| ASTACK . . . . . | 0018   | PARA  |         | STACK |
| CODE . . . . .   | 009B   | PARA  |         | NONE  |
| DATA . . . . .   | 000E   | PARA  |         | NONE  |

Symbols:

| N a m e       | Type   | Value | Attr |
|---------------|--------|-------|------|
| A . . . . .   | L WORD | 0000  | DATA |
| B . . . . .   | L WORD | 0002  | DATA |
| F12 . . . . . | L NEAR | 0009  | CODE |



|                     |             |      |               |
|---------------------|-------------|------|---------------|
| F12STEP1 . . . . .  | L NEAR 001C | CODE |               |
| F12STEP2 . . . . .  | L NEAR 0041 | CODE |               |
| F3 . . . . .        | L NEAR 005B | CODE |               |
| F3END . . . . .     | L NEAR 0097 | CODE |               |
| F3STEP1 . . . . .   | L NEAR 0066 | CODE |               |
| F3STEP2 . . . . .   | L NEAR 0071 | CODE |               |
| F3STEP3 . . . . .   | L NEAR 0082 | 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 = 009B |
| MAX4 . . . . .      | L NEAR 0094 | CODE |               |
|                     |             |      |               |
| RES . . . . .       | L WORD 000C | DATA |               |
|                     |             |      |               |
| @CPU . . . . .      | TEXT 0101h  |      |               |
| @FILENAME . . . . . | TEXT 1ab3   |      |               |
| @VERSION . . . . .  | TEXT 510    |      |               |

76 Source Lines

76 Total Lines

25 Symbols

48016 + 461291 Bytes symbol space free

0 Warning Errors

0 Severe Errors

