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. 0383 _____ Арсентьева. Д.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

Изучение представления и обработки целых чисел, а также организация ветвящихся процессов

Задание.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет:

a) значения функций i1 = f1(a,b,i) и i2 = f2(a,b,i);

b) 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.

Замечания:

1) при разработке программы нельзя использовать фрагменты, представленные на ЯВУ, в частности, для ввода-вывода данных. Исходные данные должны вводиться, а результаты контролироваться в режиме отладки;

2) при вычислении функций f1 и f2 вместо операции умножения следует использовать арифметический сдвиг и, возможно, сложение;

3) при вычислении функций f1 и f2 нельзя использовать процедуры;

4) при разработке программы следует минимизировать длину кода, для чего, если надо, следует преобразовать исходные выражения для вычисления функций.

Выполнение работы.

Вариант 21: Шифр задания (n1,n2,n3): 4.7.2

$$i1 = f4 = \begin{cases} -(6*i - 4), & \text{при } a>b \\ 3*(i+2), & \text{при } a<=b \end{cases}$$

$$i2 = f7 = \begin{cases} -(4*i - 5), & \text{при } a>b \\ 10 - 3*i, & \text{при } a<=b \end{cases}$$

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

Таблица 1 – Проверка различных маршрутов выполнения программы, а также различных знаков параметров a и b.

Введенные значения	Полученные значения			Вывод
	i1 (a,b,i)	i2 (a,b,i)	res (i1,i2,k)	
a = 5 b = 5 i = 7 k = 0	001B h = 27	FFF5 h = -11	0026 h = 38	Все верно
a = 10 b = 7 i = 11 k = -5	FFC2 h = -62	FFD9 h = -39	0031 h = 49	Все верно

a = 50 b = 150 i = -7 k = 100	FFF1 h = -15	001F h = 31	002E h = 46	Все верно
a = 15 b = 15 i = 8 k = -110	001E h = 30	FFF2 h = -14	001E h = 30	Все верно

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении В.

Выводы.

Были изучены представления и обработки целых чисел, а также организация ветвящихся процессов.

ПРИЛОЖЕНИЕ А
ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: Lab3.asm

```
; Стек  программы
AStack    SEGMENT  STACK
        DW 12 DUP('!')    ; Отводится 12 слов памяти
AStack    ENDS

; Данные программы
DATA      SEGMENT

;  Директивы описания данных
a      DW    5
b      DW    5
i      DW    7
k      DW    0
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       ;\  Сохранение адреса начала PSP в стеке
        sub   AX,AX    ; > для последующего восстановления по
        push  AX       ;/  команде ret, завершающей процедуру.
        mov   AX,DATA   ; Загрузка сегментного
        mov   DS,AX     ; регистра данных.

        ;вычисление функций
        mov cx, i
        shl cx, 1
        add cx, i
        mov ax, cx     ; ax = 3i
        mov cx, b
        cmp a, cx       ; сравниваем значения a и b
        jle LessEqual     ; если a <= b, то тогда переходим к f4second
```

```
        mov cx, 4     ; вычисление f4
        sub cx, ax
        sub cx, ax
        mov i1, cx
        mov cx, 5     ; вычисление f7
        sub cx, ax
        sub cx, i
        mov i2, cx
        jmp F2
    LessEqual:
        mov cx, 6     ; вычисление f4
        add cx, ax
        mov i1, cx
        mov cx, 10    ; вычисление f7
        sub cx, ax
        mov i2, cx

    F2:    ; рассчет f2 , cx = i2
    neg cx
    cmp k, 0
    jge GreaterEqual
        add cx, 10
        cmp i1, cx
        jl MainExit
        mov cx, i1    ; |i1| >= 10-i2
            jmp MainExit
    GreaterEqual:
        add cx, i1
        cmp cx, 0
        jge MainExit
            neg cx
            jmp MainExit
    MainExit:    ; в cx лежит значение функции f2
        ret      ; Выход в DOS по команде, находящейся в 1-ом слове PSP.

Main    ENDP
CODE    ENDS
    END Main
```

ПРИЛОЖЕНИЕ В
ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: Lab3.lst

```
Microsoft (R) Macro Assembler Version 5.10              11/7/21 13:09:44
                                                        Page     1-1


                        ; ミ。ムひオミコ ミソム€ ミセミウム€ ミーミシミシム・
0000                    AStack    SEGMENT  STACK
0000  000C[                       DW 12 DUP('!')    ; ミ樮ひイミセミエミクムび・
                        ・12 ム・サミセミイ ミソミ−ミシム肖ひク
        0021
                    ]

0018                    AStack    ENDS

                        ; ミ頒−ミスミスム巾オ ミソム€ ミセミウム€ ミーミシミシム・
0000                    DATA      SEGMENT

                        ; ミ頒クム€ ミオミコムひクミイム・ミセミソミクム・−ミスミクム・ミエミ−ミスミス
                        ム錦・
0000  0005              a      DW    5
0002  0005              b      DW    5
0004  0007              i      DW    7
0006  0000              k      DW    0
0008  0000              i1     DW    0
000A  0000              i2     DW    0
000C                    DATA      ENDS

                        ; ミ墟セミエ ミソム€ ミセミウム€ ミーミシミシム・
0000                    CODE      SEGMENT
                              ASSUME CS:CODE, DS:DATA, SS:AStack

                        ; ミ寅セミサミセミイミスミ−ム・ミソム€ ミセム・オミエム ┬€ ミ−
0000                    Main      PROC  FAR
```

```
0000  1E                        push  DS      ;\ ミ。ミセム・€ ミ−ミスミオミスミク
                                ミオ ミ−ミエム€ ミオム・− ミスミ−ム・−ミサミ− PSP ミイ ム・ひオミコミオ
0001  2B C0                     sub   AX,AX   ; > ミエミサム・ミソミセム・サミオミ
                                エム τ 紗禍オミウミセ ミイミセム・・ひ−ミスミセミイミサミオミスミクム・ミソミセ
0003  50                        push  AX      ;/ ミコミセミシミ−ミスミエミオ ret
                                , ミキミ−ミイミオム€ ム威−ム紗禍オミケ ミソム€ ミセム・オミエム τ€ ム・
0004  B8 ---- R                 mov   AX,DATA ; ミ厘−ミウム€ ム σ キミコミ− ム・
                                ミオミウミシミオミスムひスミセミウミセ
0007  8E D8                     mov   DS,AX   ; ム€ ミオミウミクム・び€ ミ− ミエ
                                ミ−ミスミスム錦・

                                ;ミイム錦・クム・サミオミスミクミオ ム・σ スミコム・クミケ
0009  8B 0E 0004 R                 mov cx, i
000D  D1 E1                        shl cx, 1
000F  03 0E 0004 R                 add cx, i
0013  8B C1                        mov ax, cx    ; ax = 3i
0015  8B 0E 0002 R                 mov cx, b
0019  39 0E 0000 R                 cmp a, cx     ; ム・€ ミ−ミイミスミクミイミ−ミオミ
                                シ ミキミスミ−ム・オミスミクム・a ミク b
001D  7E 1B                        jle LessEqual ; ミオム・サミク a <= b,
                                ムひセ ムひセミウミエミ− ミソミオム€ ミオム・セミエミクミシ ミコ f4second
001F  B9 0004                      mov cx, 4     ; ミイム錦・クム・サミオミスミ
                                クミオ f4
0022  2B C8                        sub cx, ax
0024  2B C8                        sub cx, ax
0026  89 0E 0008 R                 mov i1, cx
```



```
002A  B9 0005                      mov cx, 5     ; ミイム錦・クム・サミオミスミ
                                クミオ f7
```

```
002D  2B C8                     sub cx, ax
002F  2B 0E 0004 R              sub cx, i
0033  89 0E 000A R              mov i2, cx
0037  EB 13 90                  jmp F2
003A                        LessEqual:
003A  B9 0006                   mov cx, 6     ; ﾐｲﾑ錦･ｸﾑ･ｻﾐｵﾐｽﾐ
                            ｸﾐｵ f4
003D  03 C8                     add cx, ax
003F  89 0E 0008 R              mov i1, cx
0043  B9 000A                   mov cx, 10    ; ﾐｲﾑ錦･ｸﾑ･ｻﾐｵﾐｽﾐ
                            ｸﾐｵ f7
0046  2B C8                     sub cx, ax
0048  89 0E 000A R              mov i2, cx

004C                        F2:   ; ﾑ€ ﾐ−ﾑ･･･ｵﾑ･f2 , cx = i2
004C  F7 D9                     neg cx
004E  83 3E 0006 R 00           cmp k, 0
0053  7D 10                     jge GreaterEqual
0055  83 C1 0A                   add cx, 10
0058  39 0E 0008 R               cmp i1, cx
005C  7C 15                      jl MainExit
005E  8B 0E 0008 R               mov cx, i1    ; |i1| >= 10-i2
0062  EB 0F 90                       jmp MainExit
0065                        GreaterEqual:
0065  03 0E 0008 R               add cx, i1
0069  83 F9 00                   cmp cx, 0
006C  7D 05                      jge MainExit
006E  F7 D9                          neg cx
0070  EB 01 90                       jmp MainExit
0073                        MainExit:   ; ﾐｲ cx ﾐｻﾐｵﾐｶﾐｸﾑ･ﾐｷﾐｽﾐ
                            −ﾑ･ｵﾐｽﾐｸﾐｵ ﾑ･σ ｽﾐｺﾑ･ｸﾐｸ f2
0073  CB                        ret     ; ﾐ柘錦･ｾﾐｴ ﾐｲ DOS ﾐｿﾐｾ
                            ﾐｺﾐｾﾐｼﾐ−ﾐｽﾐｴﾐｵ, ﾐｽﾐ−ﾑ･ｾﾐｴﾑ肖禍ｵﾐｹﾑ･･ﾐｲ 1-ﾐｾ
                            ﾐｼ ﾑ･ｻﾐｾﾐｲﾐｵ PSP.

0074                        Main      ENDP
0074                        CODE      ENDS
                                END Main
```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	0074	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	004C	CODE	
GREATEREQUAL	L NEAR	0065	CODE	
I	L WORD	0004	DATA	
I1	L WORD	0008	DATA	
I2	L WORD	000A	DATA	
K	L WORD	0006	DATA	
LESSEQUAL	L NEAR	003A	CODE	
MAIN	F PROC	0000	CODE	Length = 0074
MAINEXIT	L NEAR	0073	CODE	
@CPU	TEXT	0101h		
@FILENAME	TEXT	lab3		
@VERSION	TEXT	510		

75 Source Lines

```
   75 Total   Lines
   19 Symbols

48056 + 459204 Bytes symbol space free

    0 Warning Errors
    0 Severe  Errors
```