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
Вариант 19

Студентка гр. 1383 _____ Седова Э.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучение понятия ветвящихся процессов. Разработка программы, обрабатывающей целые числа, на языке Ассемблера.

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.

Выполнение работы.

$$I1 = f4 = \begin{cases} -(6*i - 4), & \text{при } a>b \\ 3*(i+2), & \text{при } a<=b \end{cases}$$

$$I2 = f5 = \begin{cases} 20 - 4*i, & \text{при } a>b \\ -(6*I - 6), & \text{при } a<=b \end{cases}$$

$$Res = f7 = \begin{cases} |i1| + |i2|, & \text{при } k<0 \\ \max(6, |i1|), & \text{при } k>=0 \end{cases}$$

Таблица 1 – примеры тестовых случаев

Номер	Входные данные	Выходные данные
1	a=5 b=8 i=2	i1=000C =12 i2= FFFA=-6 res = 0034=12

	k=3	
2	a=2 b=-3 i=-8 k=-1	i1=0034=52 i2=0034=52 res = 0034=104
3	a=-40 b=0 i=1 k=1	i1=0009=9 i2=0000=0 res = 0009=9
4	a=0 b=0 i=0 k=0	i1=0006=6 i2=0006=6 res = 0006=6

Для минимализации длины кода были произведены следующие упрощения:

1. При a>b:

-(6i – 4) = -((2i + i)2 – 4)

20 – 4i = -4(i – 5)

2. При a<=b:

Пусть 3(i+2) = x, тогда -(6i - 6)= 2(9 – x)

3. Так как в f3 всегда используется модуль f1, метка getabs_i1, необходимая для взятия модуля от f1, вынесена отдельно. Это помогает избежать дублирования кода.

Программный код см. в приложении А.

Файлы диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения работы было изучено понятие ветвящихся процессов. Разработана программа, на языке Ассемблера, обрабатывающая целые числа.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 3lb.asm

```
      DOSSEG                                          ; Задание сегментов под
ДОС
         .MODEL   SMALL                                  ; Модель памяти-
SMALL(Малая)
         .STACK   100h                                 ; Отвести под Стек
256 байт
         .DATA                                          ; Начало сегмента
данных
           a dw 2
           b dw -3
           i dw -8
           k dw -1
           i1 dw 0
           i2 dw 0
           res dw 0
         .CODE                                     ; Начало сегмента кода
         mov   ax, @data                               ; Загрузка в DS адреса
начала
         mov   ds, ax                               ; сегмента данных

      mov ax, a
      cmp ax, b  ;сравниваем равны ли a и b.
      jle second      ;выполняет короткий переход, если первый операнд >
второго операнда

      first:          ;if(a>b)
      mov cx, i  ;i
      shl cx, 1  ;2i
      add cx, i  ;3i
      shl cx, 1  ;6i
      sub cx, 4  ;6i-4
      neg cx

      mov ax, i  ;i
      sub ax, 5  ;i-5
      shl ax, 1  ;2i-10
      shl ax, 1  ;4i-20
      neg ax          ;20-4i
      jmp result

      second:            ;if(a<=b)
      mov cx, i  ;i
      add cx, i  ;2i
      add cx, i  ;3i
      add cx, 6  ;3i+6

      mov ax, cx ;3i+6
      neg ax             ;-3i-6
      add ax, 9  ;-3i+3
      shl ax, 1  ;-6i+6
```

```
result:
     mov [i1], cx
     mov [i2], ax

     mov bx, k
     cmp bx, 0
getabs_i1:
     neg cx
     js getabs_i1
jge final2 ;короткий переход, если первый операнд >= второго
операнда
final1:
     getabs_i2:
          neg ax
          js getabs_i2
     add cx, ax
     jmp answer

final2:
     cmp cx, 6
     jge answer
     mov cx, 6
answer:
     mov [res], cx
     mov ah, 4ch
     int 21h                              ; завершение программы и
выход в ДОС
     END
```

ПРИЛОЖЕНИЕ Б
ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: 3lb.lst

```
Microsoft (R) Macro Assembler Version 5.10 11/5/22 00:11:26
                                                        Page 1-1


                        DOSSEG                                  ;
                       P—P°PrP°PSPëPµ   CЃPµPiPjPµPSC,PsPI   PïPsPґ
P”PћPЎ
                          .MODEL  SMALL
                       ; PњPsPґPµP»CЊ PïP°PjCЏC,Pë-SMALL(PњP°P»P°CЏ)
                          .STACK  100h
                       ; PћC,PIPµCЃC,Pë PïPsPґ PЎC,PµPє 256 P±P°P№C,
                          .DATA
                       ; PќP°C‡P°P»Ps CЃPµPiPjPµPSC,P° PrP°PSPSC‹C…
0000  0005                      a dw 5
0002  0008                      b dw 8
0004  0002                      i dw 2
0006  0003                      k dw 3
0008  0000                      i1 dw 0
000A  0000                      i2 dw 0
000C  0000                      res dw 0
                          .CODE                                 ;
PќP°C
                       ‡P°P»Ps CЃPµPiPjPµPSC,P° PєPsPґP°
0000  B8 ---- R         mov  ax, @data                  ; P—P°
                       PiCЂCґP·PєP° PI DS P°PґCЂPµCЃP° PSP°C‡P°P»P°
0003  8E D8                    mov                             ds,
ax                       ; CЃPµ
                       PiPjPµPSC,P° PrP°PSPSC‹C…

0005  A1 0000 R        mov ax, a
```

```
      0008  3B 06 0002 R        cmp ax, b  ;CЃCЂP°PIPSPëPIP°PµPj
CЂP°PIPSC
                          ‹ P»Pë a Pë b.
      000C  7E 20               jle second      ;PIC‹ PïPsP»PSCЏPµC,
PєPsCЂPsC,P
                          єPëP№  PïPµCЂPµC…PsPґ,  PµCЃP»Pë  PïPµCЂPIC‹ P№
PsP
                          ïPµCЂP°PSPґ > PIC,PsCЂPsPiPs PsPïPµCЂP°PSPґP°

      000E                first:          ;if(a>b)
      000E  8B 0E 0004 R        mov cx, i  ;i
      0012  D1 E1               shl cx, 1  ;2i
      0014  03 0E 0004 R        add cx, i  ;3i
      0018  D1 E1               shl cx, 1  ;6i
      001A  83 E9 04            sub cx, 4  ;6i-4
      001D  F7 D9               neg cx

      001F  A1 0004 R     mov ax, i  ;i
      0022  2D 0005             sub ax, 5  ;i-5
      0025  D1 E0               shl ax, 1  ;2i-10
      0027  D1 E0               shl ax, 1  ;4i-20
      0029  F7 D8               neg ax          ;20-4i
      002B  EB 19 90            jmp result

      002E                second:            ;if(a<=b)
      002E  8B 0E 0004 R        mov cx, i  ;i
      0032  03 0E 0004 R        add cx, i  ;2i
      0036  03 0E 0004 R        add cx, i  ;3i
      003A  83 C1 06            add cx, 6  ;3i+6

      003D  8B C1               mov ax, cx ;3i+6
      003F  F7 D8               neg ax             ;-3i-6
```

```
0041  05 0009            add ax, 9  ;-3i+3
0044  D1 E0              shl ax, 1  ;-6i+6

0046                result:
0046  89 0E 0008 R            mov [i1], cx
004A  A3 000A R          mov [i2], ax

004D  8B 1E 0006 R            mov bx, k
0051  83 FB 00                cmp bx, 0
0054                getabs_i1:
0054  F7 D9                   neg cx
0056  78 FC                   js getabs_i1
0058  7D 09              jge final2 ;РєРѕСЂРѕС‚РєРёР№
РїРµСЂРµС…РѕР
                      ґ,  РµСЃР»Рё  РїРµСЂРІС‹Р№  РѕРїРµСЂР°РЅРґ  >=
PIC,
                      РѕСЂРіРіРѕ РѕРїРµСЂР°РЅРґР°
005A                final1:
005A                     getabs_i2:
005A  F7 D8                        neg ax
005C  78 FC                        js getabs_i2
005E  03 C8                   add cx, ax
0060  EB 09 90                jmp answer

0063                final2:
0063  83 F9 06                cmp cx, 6
0066  7D 03                   jge answer
0068  B9 0006                 mov cx, 6
006B                answer:
006B  89 0E 000C R            mov [res], cx
006F  B4 4C                   mov ah, 4ch
0071  CD 21                   int 21h                    ;
Р·Р°
```

```
                        PIPµCЂC€PµPSPëPµ     PïCЂPsPiCЂP°PjPjC‹     Pë
PIC‹C…Ps
                        Pґ PI P”PћPЎ
                             END
     Microsoft      (R)      Macro      Assembler      Version      5.10
11/5/22 00:11:26

Symbols-1


     Segments and Groups:

                N a m e         Length     Align     Combine Class

     DGROUP . . . . . .         GROUP
       _DATA  . . . . . .       000E  WORD  PUBLIC     'DATA'
       STACK  . . . . . .       0100  PARA  STACK 'STACK'
     _TEXT  . . . . . .         0073  WORD  PUBLIC     'CODE'

     Symbols:

                N a m e         Type  Value      Attr

     A  . . . . . .             L WORD     0000  _DATA
     ANSWER . . . . . .         L NEAR     006B  _TEXT

     B  . . . . . .             L WORD     0002  _DATA

     FINAL1 . . . . . .         L NEAR     005A  _TEXT
     FINAL2 . . . . . .         L NEAR     0063  _TEXT
     FIRST  . . . . . .         L NEAR     000E  _TEXT

     GETABS_I1  . . . . . .     L NEAR     0054  _TEXT
     GETABS_I2  . . . . . .     L NEAR     005A  _TEXT

     I  . . . . . .             L WORD     0004  _DATA
     I1 . . . . . .             L WORD     0008  _DATA
     I2 . . . . . .             L WORD     000A  _DATA
```

```
K  . . . . . .  L WORD    0006 _DATA

RES  . . . . . .  L WORD    000C _DATA
RESULT . . . . . .  L NEAR    0046 _TEXT

SECOND . . . . . .  L NEAR    002E _TEXT

@CODE  . . . . . .  TEXT  _TEXT
@CODESIZE  . . . . . .  TEXT  0
@CPU . . . . . .  TEXT  0101h
@DATASIZE  . . . . . .  TEXT  0
@FILENAME  . . . . . .  TEXT  _3lb
@VERSION . . . . . .  TEXT  510
```

Microsoft (R) Macro Assembler Version 5.10 11/5/22
00:11:26

Symbols-2

73 Source Lines
73 Total Lines
32 Symbols

48004 + 457206 Bytes symbol space free

0 Warning Errors
0 Severe Errors