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»

Студент гр. 1383 _____ Петров А.С.

Преподаватель _____ Ефремов М. А.

Санкт-Петербург
2022

Цель работы.

Изучить представление и обработку целых целых чисел, а также организацию ветвящихся процессов.

Задание на лабораторную работу.

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.

Входные данные (вариант 16)

$$i1 = \begin{cases} 7 - 4*i, & a>b \\ 8 - 6*i, & a<=b \end{cases}$$

$$i2 = \begin{cases} 2 (i + 1) - 4, & a>b \\ 5 - 3*(i + 1), & a<=b \end{cases}$$

$$res = \begin{cases} \min(|i1 - i2|, 2), & k<0 \\ \max(-6, -i2), & k>=0 \end{cases}$$

Выполнение работы.

Были произведены следующие упрощения:

1. $2*(i + 1) – 4 = 2*i – 2$
2. $5 – 3*(i + 1) = 2 – 3*i$

Упрощения были произведены для уменьшения выполняемых операций и следовательно уменьшению кода программы.

Тестирование

Результаты тестирования представлены в таблице 1.

Таблица 1 – Результаты тестирования.

Номер теста	Входные данные	Результат	Ожидаемый результат
1	a = 1 b = 3 i = -1 k = 4	i1 = 14 i2 = 5 res = -5	i1 = 14 i2 = 5 res = -5
2	a = -1 b = 2 i = 1 k = 3	i1 = 2 i2 = -1 res = 1	i1 = 2 i2 = -1 res = 1
3	a = 6 b = -3 i = -2 k = 0	i1 = 15 i2 = -6 res = 6	i1 = 15 i2 = -6 res = 6
4	a = -2 b = -5 i = 2 k = -1	i1 = -1 i2 = 2 res = 2	i1 = -1 i2 = 2 res = 2

Выводы.

В ходе выполнения работы были изучены ветвящиеся процессы и представление целых чисел.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr3.asm

```
; Стек  программы
AStack    SEGMENT  STACK
          DW 12 DUP(?)
AStack    ENDS

; Данные программы
    DATA      SEGMENT

    a       DW    -2
    b       DW    5
    i       DW    1
    k       DW    -1
    i1      DW    0
    i2      DW    0
    res     DW    0
    DATA      ENDS

; Код программы
CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
Main      PROC  FAR
    push  DS
    sub   AX,AX
    push  AX
    mov   AX,DATA
    mov   DS,AX

i_1_0:
    mov cx, a
    mov dx, b
    cmp dx, cx
    jge i_1_2; if(b>=a)

i_1_1:
    mov ax, i
    shl ax, 1; 2i
    shl ax, 1; 4i
    neg ax; -4i
    add ax, 7; -4i+7
    mov i1, ax
    jmp i_2_0

i_1_2:
    mov ax, i
```

```
    shl ax, 1; 2i
    add ax, i; 3i
    shl ax, 1; 6i
    neg ax; -6i
    add ax, 8
    mov i1, ax; -6i+8

i_2_0:
    cmp dx, cx
    jge i_2_2; if(b>=a)

i_2_1:
    mov ax, i
    shl ax, 1; 2i
    sub ax, 2; 2i-2
    jmp i_2_end

i_2_2:
    mov ax, i
    shl ax, 1; 2i
    add ax, i; 3i
    neg ax; -3i
    add ax, 2; 2-3i

i_2_end:
    mov i2, ax

res_0:
    mov cx, k
    mov dx, 0
    cmp cx, dx
    jge res_2; if(k>=0)

res_1:
    mov ax, i1
    mov bx, i2
    sub ax, bx; i1-i2
    cmp ax, 0
    jge res_1_1; if(i1-i2>=0)

res_1_abs:
    neg ax;|i1-i2|

res_1_1:
    cmp ax, 2
    jge res_1_end; if(|i1-i2|>=2)

    mov res, ax;|i1-i2|
    push ax
    jmp en

res_1_end:
    mov res, 2
```

```
    push [res]
    jmp en

res_2:
    mov ax, i2
    neg ax; -i2
    mov bx, -6
    cmp ax, bx
    jle res_2_2; if(-i2<=-6)

res_2_1:
    mov res, ax
    push ax
    jmp en

res_2_2:
    mov res, bx
    push bx
    jmp en

en:
    ret

Main      ENDP
CODE      ENDS
          END Main

; Стек  программы
AStack    SEGMENT  STACK
          DW 12 DUP(?)
AStack    ENDS

; Данные программы
    DATA      SEGMENT

    a       DW    1
    b       DW    3
    i       DW    -1
    k       DW    4
    i1      DW    0
    i2      DW    0
    res     DW    0
    DATA      ENDS

; Код программы
CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
Main      PROC  FAR
    push  DS
    sub   AX,AX
    push  AX
```

```
    mov   AX,DATA
    mov   DS,AX

i_1_0:
    mov cx, a
    mov dx, b
    cmp dx, cx
    jle i_1_2; if(a>=b)

i_1_1:
    mov ax, i
    shl ax, 1; 2i
    add ax, i; 3i
    shl ax, 1; 6i
    neg ax; -6i
    add ax, 8
    mov i1, ax; -6i+8
    jmp i_2_0

i_1_2:
    mov ax, i
    shl ax, 1; 2i
    shl ax, 1; 4i
    neg ax; -4i
    add ax, 7; -4i+7
    mov i1, ax

i_2_0:
    cmp dx, cx
    jle i_2_1

i_2_2:
    mov ax, i
    shl ax, 1; 2i
    add ax, i; 3i
    neg ax; -3i
    add ax, 2; 2-3i
    jmp i_2_end

i_2_1:
    mov ax, i
    shl ax, 1; 2i
    sub ax, 2; 2i-2

i_2_end:
    mov i2, ax

res_0:
    mov cx, k
    mov dx, 0
    cmp cx, dx
    jle res_1
```

```
res_2:
    mov ax, i2
    neg ax; -i2
    mov bx, -6
    cmp ax, bx
    jle res_2_2

res_2_1:
    mov res, ax
    push ax
    jmp en

res_2_2:
    mov res, bx
    push bx
    jmp en

res_1:
    mov ax, i1
    mov bx, i2
    sub ax, bx
    cmp ax, 0
    jge res_1_1; if(ax>=0)

res_1_abs:
    neg ax

res_1_1:
    cmp ax, 2
    jle res_1_end

    mov res, ax
    push ax
    jmp en

res_1_end:
    mov res, 2
    push [res]

en:
    ret

Main      ENDP
CODE      ENDS
          END Main
```