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. 1303 Иевлев Е.А.

Преподаватель Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров вычисляет значения функций.

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.

| 8 | 2.3.8 |

$$f2 = \begin{cases} -(4*i+3), & \text{при } a>b \\ 6*i-10, & \text{при } a<=b \end{cases}$$

$$f3 = \begin{cases} 7-4*i, & \text{при } a>b \\ 8-6*i, & \text{при } a<=b \end{cases}$$

$$f8 = \begin{cases} |i1|-|i2|, & \text{при } k<0 \\ \max(4,|i2|-3), & \text{при } k>=0 \end{cases}$$

Выполнение работы

1. Из таблицы получен вариант набора функций, приведенного в каталоге задания, которые необходимо реализовать.

2. Программа протранслирована с различными значениями переменных, результат выполнения набора функций зафиксирован в таблице;

Для выполнения данного задания были использованы такие команды общего назначения как:

Команды передачи данных.

1) Mov – присваивание

Двоичные арифметические команды.

1) Add - сложение
2) Sub - вычитание
3) Cmp – сравнение
4) Neg – смена знака

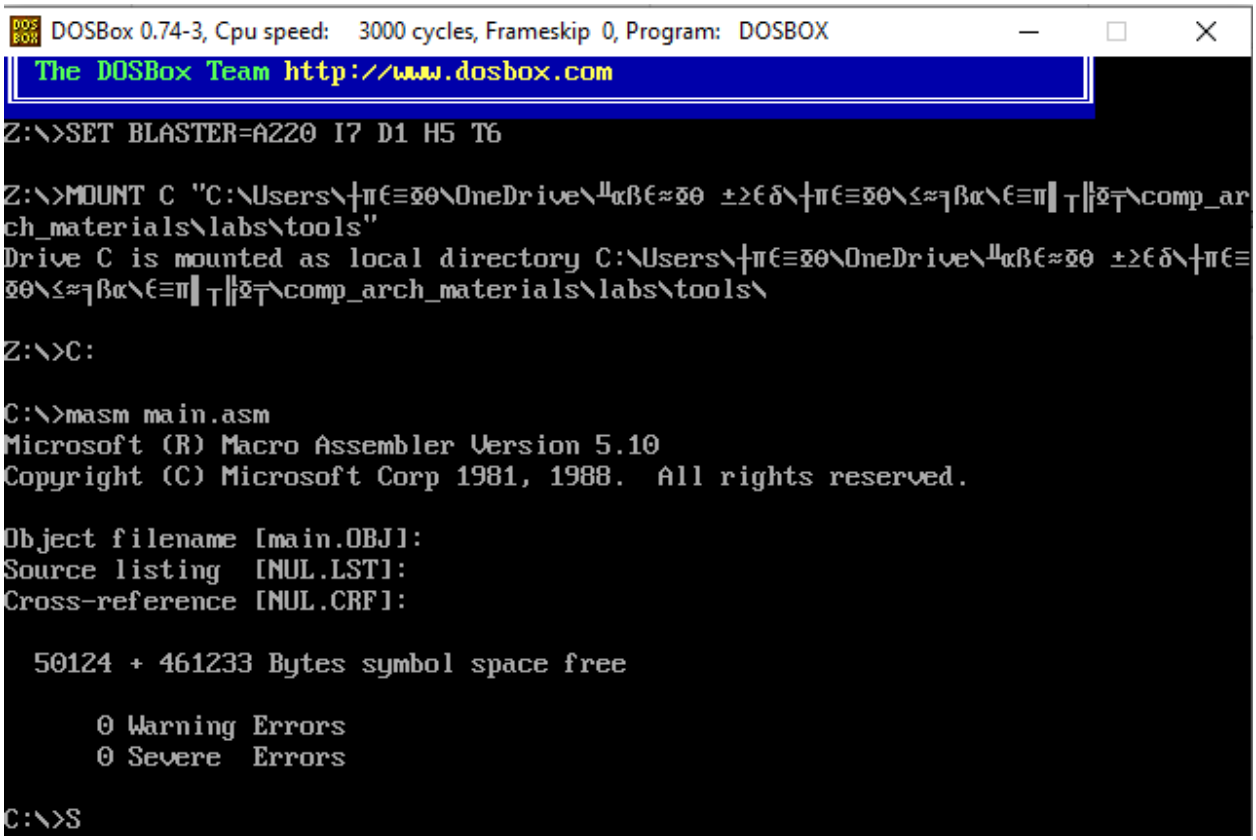
Команды побитового сдвига.

1) Shl - арифметический сдвиг влево

Команды передачи управления.

1) Jmp - безусловный переход
2) Int - вызов программного прерывания
3) Jge(jump greater equal) - выполняет короткий переход, если первый операнд больше второго операнда или равен ему при выполнении операции сравнения с помощью команды cmp
4) Jg(jump greater) - выполняет короткий переход, если первый операнд больше второго операнда при выполнении операции сравнения с помощью команды cmp.
5) Jle(jump less than or equal) - выполняет короткий переход, если первый операнд меньше или равен второму операнду при выполнении операции сравнения с помощью команды cmp.

Также были использованы метки (для примера B1, C2), для перехода между некоторыми командами. Метка - это символьное имя, обозначающее ячейку памяти, которая содержит некоторую команду.



Трансляция программы

4. Программа выполнена в пошаговом режиме под управлением отладчика с фиксацией значений используемых переменных.

source.exe

№ теста	Тестируемый случай	Функции для данного случая	Данные	
			входные	выходные
1	a > b k < 0	f2 = -(4i + 3) f3 = 7 – 4i f8 = |i1| - |i2|	a = 2, b = 1 k = -1 i = 1	f2 = -7 = FFF9 f3 = 3 = 0003 f8 = 4 = 0004
2	a > b k >= 0	f2 = -(4i + 3) f3 = 7 – 4i f8 = max(4, |i2| - 3)	a = 2, b = 1 k = 0 i = 1	f2 = -7 = FFF9 f3 = 3 = 0003 f8 = 4 = 0004
3	a > b k >= 0	f2 = -(4i + 3) f3 = 7 – 4i f8 = max(4, |i2| - 3)	a = 2, b = 1 k = 0 i = 4	f2 = -19 = FFED f3 = -9 = FFF7 f8 = 6 = 0006
4	a <= b k < 0	f2 = 6*i – 10 f3 = 8 – 6i f8 = |i1| - |i2|	a = 1, b = 2 k = -1 i = 1	f2 = -4 = FFFC f3 = 2 = 0002 f8 = 2 = 0002
5	a <= b k >= 0	f2 = 6*i – 10 f3 = 8 – 6i f8 = max(4, |i2| - 3)	a = 1, b = 2 k = 0 i = 1	f2 = -4 = FFFC f3 = 2 = 0002 f8 = 4 = 0004

Выводы

В ходе выполнения лабораторной работы были получены навыки разработки программы с заданными целочисленными значениями на языке программирования Ассемблер.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.asm

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

AStack    SEGMENT STACK
DW 32 DUP(0)
AStack    ENDS


DATA SEGMENT

a DW ?
b DW ?
k DW ?      ;f2
i DW ?      ;f3
i1 DW ?     ;f8
i2 DW ?
res DW ?

DATA ENDS

CODE SEGMENT

Main PROC FAR
push ds
sub ax, ax
push ax

mov ax, DATA
mov ds, ax
```

```
mov ax, a
mov bx, i
cmp ax, b
jle A1          ; a <= b, go to A1

mov cl, 2       ; a > b
shl bx, cl      ; 4 * i
neg bx          ; -4 * i
sub bx, 3       ; -4 * i - 3 = -(4i + 3)
mov i1, bx      ; result of f2

add bx, 10      ; 7 - 4 * i
mov i2, bx      ; result of f3
jmp C2

A1:                     ; a <= b
shl bx, 1       ; 2i
add bx, i       ; 3i
shl bx, 1       ; 6i
sub bx, 10      ; 6i - 10
mov i1, bx      ; result of f2

add bx, 2       ; 6i - 8
neg bx          ; 8 - 6i
mov i2, bx      ; result of f3

C2:
cmp i2, 0
jge B1          ; i2 >= 0, go to B1

neg bx          ; i2
B1:
mov ax, k
cmp ax, 0
jge B2          ; k >= 0, go to B2
```

```
mov cx, i1
cmp cx, 0
jge C1          ; i1 >= 0, go to C1

neg cx        ; i1

C1:
sub cx, bx    ; i1 - i2
mov res, cx   ; res = cx
jmp Exit

B2:
sub bx, 3     ; |i2| - 3
cmp bx, 4
jg B3         ; |i2| - 3 > 4

mov res, 4    ; res = 4
jmp Exit

B3:
mov res, bx   ; res = |i2| - 3

Exit:
int 20h

Main ENDP
CODE ENDS
END Main
```