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 системы»

Тема: Представление и обработка целых чисел. Организация ветвящихся процессов

Вариант 19

Студента гр. 1383 _____ Сардинов М. Г.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучение алгоритма построения оператора ветвления на языке ассемблера и методов взаимодействия с целыми числами и их обработки.

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.

1) формулу -(6i-4) можно представить в виде -6i+4
2) формулу 3*(i+2) можно представить в виде 3i+6
3) формулу 20-4i оптимизировать нет необходимости
4) формулу -(6i-6) можно представить в виде -6i+6
5) формулу |i1| + |i2| оптимизировать нет необходимости
6) формулу max(6,|i1|) оптимизировать нет необходимости

Тесты:

Номер	Входные данные	Результат
1	a = 5 b = 8	i1 = -9 i2 = 36

	k = 0 i = -5	res = 9
2	a = 8 b = 5 k = -1 i = 2	i1 = -8 i2 = 12 res = 20
3	a = 18 b = 18 k = 151 i = -31	i1 = FFA9 i2 = 00C0 res = 57
4	a = -3 b = -8 k = 0 i = 45	i1 = FEF6 i2 = FF60 res = 010A

Выводы.

В ходе работы были изучены ветвления и обработка с целых чисел на языке ассемблер.

Исходный код

```
a EQU -3
b EQU -8
k EQU 0
i EQU 45

;i1
; / -(6*i - 4) , при a>b
; f4 = <
; \ 3*(i+2) , при a<=b

;i2
; / 20 - 4*i , при a>b
; f5 = <
; \ -(6*I - 6), при a<=b

;res
; /|i1| + |i2|, при k<0
; f7 = <
; \ max(6, |i1|), при k>=0

AStack SEGMENT STACK
DW 12 DUP(?)
AStack ENDS

DATA SEGMENT
   i1 dw 0
   i2 dw 0
   res dw 0
DATA ENDS

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

mov AX, a
mov BX, b
CMP AX,BX
JG a_bigger_b
JLE b_bigger_a
```

```
a_bigger_b:
    mov ax,i
    mov cx, -6
    mul cx
    add ax, 4
    mov i1,ax

    mov ax,i
    mov cx, -4
    mul cx
    add ax, 20
    mov i2,ax
    jmp con2

b_bigger_a:
    mov ax,i
    mov cx, 3
    mul cx
    add ax, 6
    mov i1,ax

    mov ax,i
    mov cx, -6
    mul cx
    add ax, 6
    mov i2,ax

con2:
mov AX, k
mov BX, 0
CMP AX,BX
JGE k_big
JL k_small

k_small:
    mov ax, i1
    test ax, ax
    jns cont1
    neg ax
    jmp cont1

cont1:
    mov i1,ax
    mov ax, i2
    test ax, ax
    jns cont2
    neg ax
    jmp cont2
```

```
cont2:
    add ax,i1
    mov res,ax
    ret

k_big:
    mov ax, i1
    test ax, ax
    jns cont3
    neg ax
    jmp cont3

cont3:
    mov bx,6
    CMP bx,ax
    JAE a1
    mov res,ax
    ret
a1:
    mov res, 6
    ret

Main ENDP
CODE ENDS
END Main
```