**Лабораторная работа №1**

Выполнил студент группы №3145

Запорожец Арсений

**Задание №1**

**Выражение для Задачи: (a + b/c)**

**Входные значения: {{23, 28, 8}, {-19, -17, -23}}**

**Идея:**

Получим в регистры числа, после чего выясняем, есть ли среди делимого и делителя отрицательные числа. Если есть – применяем операцию idiv, если нету – div. После выводим результат, сохранившийся в регистре, на экран.

**Код программы:**

%include "io.inc"

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

mov cl, -23; Значение переменной a

mov ax, -28; Значение переменной b

mov bl, -8; Значение переменной c

cmp bl, 0

jl negative

cmp ax, 0

jl negative

div bl

xor ah, ah

add al, cl

PRINT\_STRING 'AL contain: '

PRINT\_UDEC 4, al

ret

negative:

idiv bl

xor ah, ah

add al, cl

PRINT\_STRING 'AL contain: '

PRINT\_UDEC 4, al

ret

**Задание №2**

**Выражение для Задачи: (a + b/c)**

**Входные значения: {{23, 28, 8}, {-19, -17, -23}}**

**Идея:**

Указываем все нужны числа в переменных, после чего смотрим, есть ли отрицательные переменные. Если есть – указываем в соответствующих переменных, какие именно являются таковыми. После в регистрах стека совершаем все необходимые операции, возвращаем минусы и присваиваем ответ переменной mr.

**Код программы:**

%include "io.inc"

section .data

a dd -19; Значение переменной a

b dq -17; Значение переменной б

c dq -23; Значение переменной c

p dq 0; Переменная отрицательности числа

pa dq 0; Переменная отрицательности числа "a"

r dd 0; Результат

bm dq 0; Специальная переменная для передачи данных "b" регистры стека

cm dq 0; Специальная переменная для передачи данных "c" в регистры стека

cn dq 1; Специальная переменная для уравнивания значения "a"

mr dq 0; Сцециалная переменная, предназначенная для отрицательного a

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

mov eax, [a]

cmp eax, 0

jl minus\_a

ja cont

je cont

minus\_a:

mov eax, 1

add [pa], eax

mov eax, [a]

mov ebx, -1

mul ebx

mov [a], eax

jmp cont

cont:

mov ebx, [b]

mov ecx, [c]

cmp ebx, 0

jl module1

cmp ecx, 0

jl module2

mov eax, [b]

mov [bm], eax

mov eax, [c]

mov [cm], eax

jmp continue

module1:

cmp ecx, 0

jl module3

mov eax, 1

add [p], eax

mov eax, [b]

mov ebx, -1

mul ebx

mov [bm], eax

mov eax, [c]

mov [cm], eax

jmp continue

module2:

mov eax, 1

add [p], eax

mov eax, [c]

mov ebx, -1

mul ebx

mov [cm], eax

mov eax, [b]

mov [bm], eax

jmp continue

module3:

mov eax, [b]

mov ebx, -1

mul ebx

mov [bm], eax

mov eax, [c]

mov ebx, -1

mul ebx

mov [cm], eax

jmp continue

continue:

FLD DWORD [a]; Делаем "a" переменной типа float

FLD DWORD [cn]

FDIV

fstp DWORD [a]

FLD QWORD [bm]

FLD QWORD [cm]

FDIV

fstp DWORD [r]

mov eax, 1

cmp eax, [pa]

je final\_neg\_a

final\_neg:

FLD DWORD [mr]

FLD DWORD [r]

FSUB

fstp DWORD [mr]

ret

final\_pos:

FLD DWORD [mr]

FLD DWORD [r]

FADD

fstp DWORD [mr]

ret

final\_neg\_a:

FLD DWORD [mr]

FLD DWORD [a]

FSUB

fstp DWORD [mr]

cmp eax, [p]

je final\_neg

ja final\_pos

ret

**Задание №3**

**Код программы**

**Выражения для задания: {a(x; y), b(x; y), c(x; y)}**

**Входные значения: не заданы**

**Использованные входные значения:**

**1) a(4;9), b(7;42), c(7;91)**

**2) a(-1;6), b(-11;19), c(10;10)**

**Идея:**

Вводим все необходимые значения в переменные, после чего вычисляем квадраты расстояний между точками с помощью теоремы Пифагора, а затем сравниваем их. Вычисляем квадрат самого большого и заканчиваем программу.

**Код программы:**

%include "io.inc"

;Вариант: 7

section .data

aa dq 4; Абцисса точки A

ao dq 9; Ордината точки A

ba dq 7; Абцисса точки B

bo dq 42; Ордината точки B

ca dq 7; Абцисса точки C

co dq 91; Ордината точки C

AB dq 0; Квадрат длины AB

AC dq 0; Квадрат длины AC

BC dq 0; Квадрат длины BC

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

; Считаем расстояние между точками

; Между A и B:

; Вычисляем квадрат длины катета на оси OX

xor eax, eax

mov eax, [aa]

sub eax, [ba]

mul eax

add [AB], eax; Запишем полученное значение в переменную AB

; Вычисляем квадрат длины между точками

xor eax, eax

mov eax, [ao]

sub eax, [bo]

mul eax

add [AB], eax; Запишем квадрат длины в переменную AB

; Между A и C:

; Вычисляем квадрат длины катета на оси OX

xor eax, eax

mov eax, [aa]

sub eax, [ca]

mul eax

mov [AC], eax; Запишем полученное значение в регистр AC

; Вычисляем квадрат длины между точками

xor eax, eax

mov eax, [ao]

sub eax, [co]

mul eax

add [AC], eax; Запишем квадрат длины в переменную AC

; Между B и C:

; Вычисляем квадрат длины катета на оси OX

xor eax, eax

mov eax, [ba]

sub eax, [ca]

mul eax

mov [BC], eax; Запишем полученное значение в переменную BC

; Вычисляем квадрат длины между точками

xor eax, eax

mov eax, [bo]

sub eax, [co]

mul eax

add [BC], eax; Запишем полученное значение в переменную BC

mov eax, [AB]

mov ebx, [AC]

mov ecx, [BC]

cmp eax, ecx

jl final\_1; При AB < BC

ja final\_2; При AB > BC

je final\_2; При AB = BC

final\_1:

cmp ecx, ebx

jl Big\_AC

ja Big\_BC

je Big\_BC

final\_2:

cmp eax, ebx

jl Big\_AC

ja Big\_AB

je Big\_AB

Big\_AB:

fld DWORD [AB]

fsqrt

fstp DWORD [AB]

ret

Big\_AC:

fld DWORD [AC]

fsqrt

fstp DWORD [AC]

ret

Big\_BC:

fld DWORD [BC]

fsqrt

fstp DWORD [BC]

ret

ret