**Sam Lazrak**

**CS 330 Computer Architecture/Assembly Language**

**Programing Assignment 5**

**3/30/18**

Source Code:

%include "along32.inc"

extern scanf

extern printf

section .data

valuePromt :db "Please enter the values for A, B, C!", 0ah, 0

solution1Prompt :db "The Solution is either: ", 0

solution2Prompt :db "OR: ", 0

errorSqrtPrompt :db "ERROR! Can not take square root of a negative number!",0ah,0

errorDivPrompt :db "ERROR! Can not divide by zero!",0ah, 0

A :dq 0.0

B :dq 0.0

C :dq 0.0

solution1 :dq 0.0

solution2 :dq 0.0

negB :dq 0.0

bSquared :dq 0.0

twoA :dq 0.0

four :dq 4.0

two :dq 2.0

fourAC :dq 0.0

sqrt :dq 0.0

sum :dq 0.0

diff :dq 0.0

fmt1 db "%lf", 0

fmt2 db 10, "The number is: %1.3f", 10, 0

section .text

global main

main:

call Clrscr

mov edx,valuePromt

call WriteString

call getfloat

call getfloat

call getfloat

fstp qword [C]

fstp qword [B]

ftst

fstsw ax

sahf

je errorDiv

fstp qword [A]

Logic:

fld qword [B]

fld st0

fchs

fstp qword [negB]

fmul st0, st0

fstp qword [bSquared]

fld qword [four]

fld qword [C]

fmul st0, st1

fld qword [A]

fmul st0, st1

fst qword [fourAC]

fld qword [two]

fld qword [A]

fmul st0, st1

fstp qword [twoA]

fld qword [bSquared]

fsub st2

ftst

fstsw

sahf

jb errorSqrt

fsqrt

fst qword [sqrt]

fld qword [negB]

fadd st1

fstp qword [sum]

fld qword [negB]

fsub st1

fstp qword [diff]

fld qword [twoA]

fld qword [diff]

fdiv st1

fstp qword [solution1]

fld qword [sum]

fdiv st1

fstp qword [solution2]

fninit

Output:

mov edx, solution1Prompt

call WriteString

call Crlf

fld qword [solution2]

call putfloat

call Crlf

mov edx, solution2Prompt

call WriteString

call Crlf

fld qword [solution1]

call putfloat

call Crlf

call ExitProc

errorDiv:

mov edx, errorDivPrompt

call WriteString

call ExitProc

errorSqrt:

mov edx, errorSqrtPrompt

call WriteString

call ExitProc

getfloat:

push ebp

mov ebp, esp

sub esp, 8

lea eax, [ebp - 8]

push eax

push fmt1

call scanf

add esp, 8

; (probably be a good idea to check the return from scanf here, before proceeding...)

fld qword [ebp - 8]

mov esp, ebp

pop ebp

ret

putfloat:

push ebp

mov ebp, esp

sub esp, 8

fst qword [ebp - 8]

push fmt2

call printf

add esp, 12

mov esp, ebp

pop ebp

ret