<https://youtu.be/svBlLQl7xoA>

Part A

; Program Template (lab8pA.asm)

; Program Description:

; Author: Timothy Bryant

; Creation Date: 3/8/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

val2 = 2h ;intialize val2

val3 = 3h ;intialize val3

val4 = 5h ;intialize val4

.code

main PROC

;write your code here

;32 bit

mov eax, DWORD PTR val2 ;move val2 into eax as 32 bit

mov edx, DWORD PTR val3 ;move val3 into ebx as 32 bit

mul edx ;multiple val2 and val3

mov ebx, DWORD PTR val4 ;move val4 into ebx as 32 bit

sub ebx, 3h ;subtract 3 from val4

div ebx ;divide eax by ebx

;16 bit

mov eax, WORD PTR val2

mov edx, WORD PTR val3

mul edx

mov ebx, WORD PTR val4

sub ebx, 3h

div ebx

;8 bit

mov eax, BYTE PTR val2

mov edx, BYTE PTR val3

mul edx

mov ebx, BYTE PTR val4

sub ebx, 3h

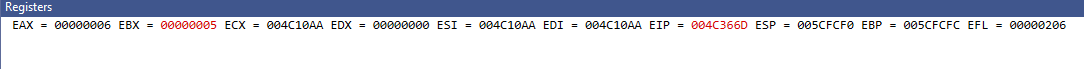
div ebx

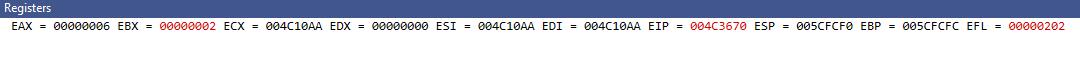
call DumpRegs

INVOKE ExitProcess,0

main ENDP

; (insert additional procedures here)

END main





Part B

; Program Template (lab8pB.asm)

; Program Description:

; Author: Timothy Bryant

; Creation Date: 3/9/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

aValue BYTE "Result of val1: ", 0 ;string for result

bValue BYTE "Enter val2: ", 0 ;string for val2 prompt

cValue BYTE "Enter val3: ", 0 ;string for val3 prompt

dValue BYTE "Enter val4: ", 0 ;string for val4 prompt

val1 DWORD ? ;intialize val1

val2 DWORD ? ;intialize val2

val3 DWORD ? ;intialize val3

val4 DWORD ? ;intialize val4

.code

main PROC

;write your code here

;prompt for val2

mov edx, OFFSET bValue

call WriteString

call ReadInt

mov val2, eax

;prompt for val3

mov edx, OFFSET cValue

call WriteString

call ReadInt

mov val3, eax

;prompt for val4

mov edx, OFFSET dValue

call WriteString

call ReadInt

mov val4, eax

;equation

mov eax, val2

mov edx, val3

mul edx

mov ebx, val4

sub ebx, 3h

div ebx

mov val1, eax

;output val1

mov edx, OFFSET aValue

call WriteString

mov eax, val1

call WriteInt

call DumpRegs

INVOKE ExitProcess,0

main ENDP

; (insert additional procedures here)

END main

