<https://youtu.be/7EN9LyH4jqg>

; Program Template (labw12pA.asm)

; Program Description: convert the high level language segment to assembly

; Author: Timothy Bryant

; Creation Date: 3/30/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

Input1 BYTE "Enter i: ", 0 ;string for input of i

Output1 BYTE "Result of f: ", 0 ;string for output of f

;intialize variables

i DWORD ?

j DWORD 25

f DWORD 1234

g DWORD 15

h DWORD 32

.code

main PROC

;write your code here

;prompt user for input of i

mov edx, OFFSET Input1

call WriteString

call ReadDec

mov i, eax

;move i into registers to compare

mov eax, i

;check if i and j are equal

cmp eax, j

jne equation2

;equation for f = g + h

mov eax, g

add eax, h

jmp bye

;equation for f = f - i

equation2: mov eax, f

sub eax, i

bye: mov edx, OFFSET Output1

call WriteString

call WriteDec

call DumpRegs

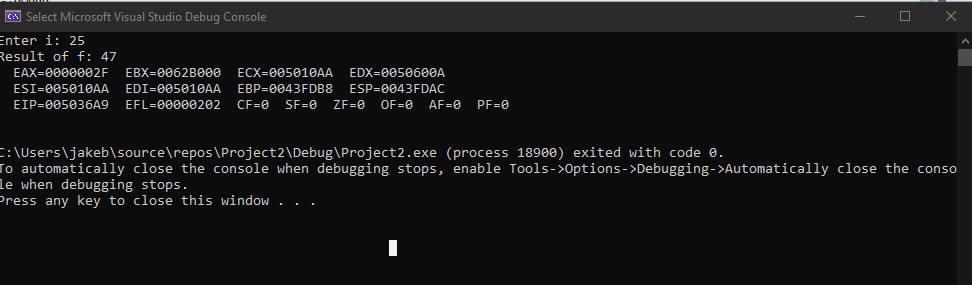
INVOKE ExitProcess, 0

main ENDP

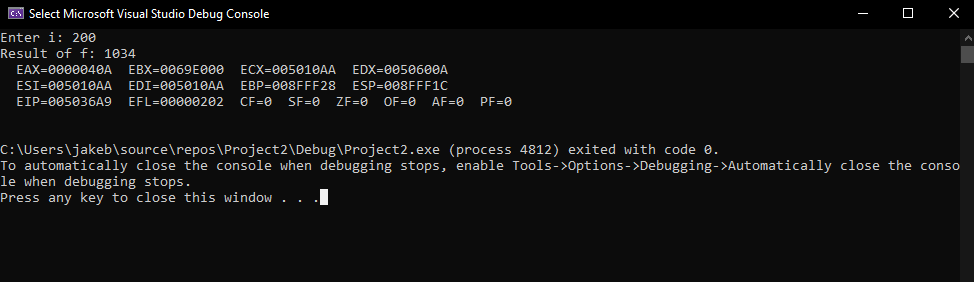
; (insert additional procedures here)

END main

I=j



I!=j



; Program Template (labw12pB.asm)

; Program Description: convert the high level language segment to assembly

; Author: Timothy Bryant

; Creation Date: 3/30/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

Input1 BYTE "Please enter a power of 2 value and I shall calculate xWhere 2 \*\* x = the value you entered: ", 0

;output for loop

Output1 BYTE "x is ", 0

Output2 BYTE ", where 2 \*\*", 0

Output3 BYTE " = ", 0

pow DWORD 1

x DWORD 0

val1 DWORD ?

.code

main PROC

;write your code here

;prompt user for 128

mov edx, OFFSET Input1

call WriteString

call ReadDec

mov val1, eax

;move values into registers

mov eax, pow

mov ebx, x

mov ecx, 2

mov esi, val1

;loop

restart:

mul ecx ;pow\*2

add ebx, 1 ;x + 1

cmp eax, esi ;comparison of pow and val1

jne restart

;output

mov edx, OFFSET Output1

call WriteString

mov eax, ebx ;move value of x after loop in eax

call WriteDec

mov edx, OFFSET Output2

call WriteString

call WriteDec

mov edx, OFFSET Output3

call WriteString

mov eax, val1

call WriteDec

call DumpRegs

INVOKE ExitProcess, 0

main ENDP

; (insert additional procedures here)

END main

