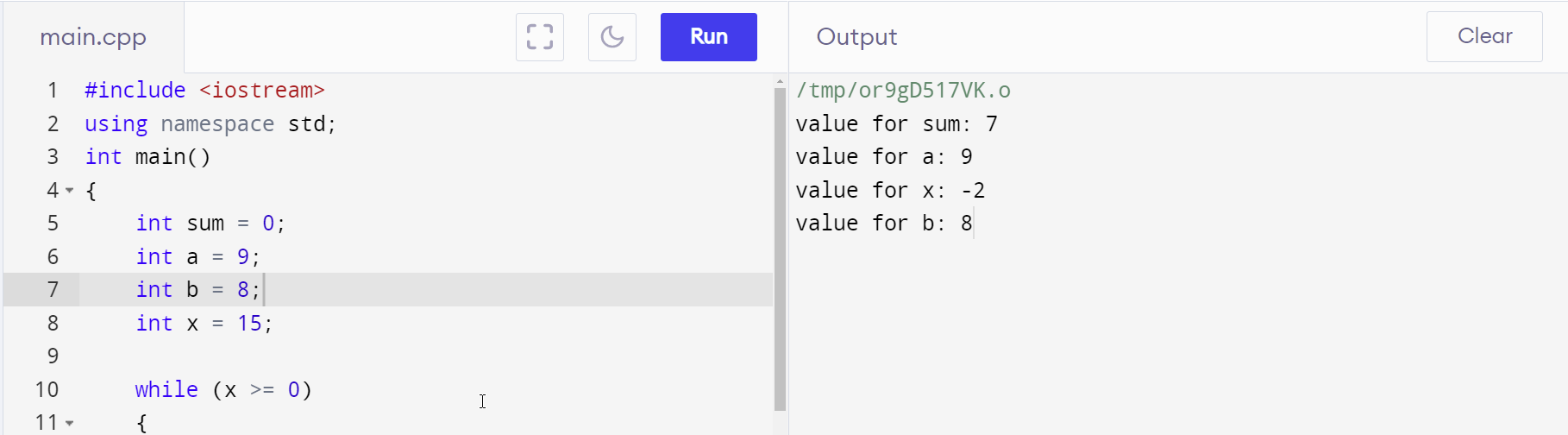
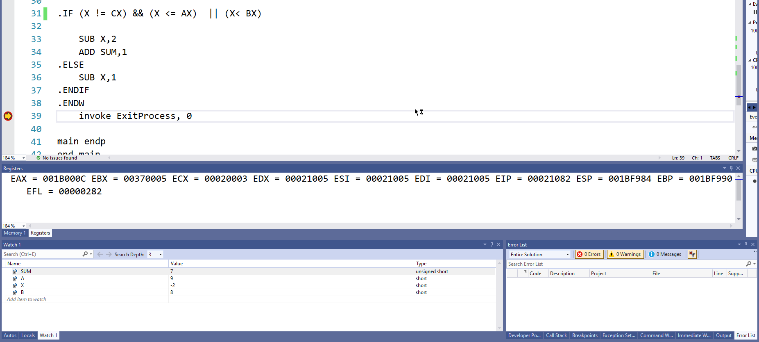
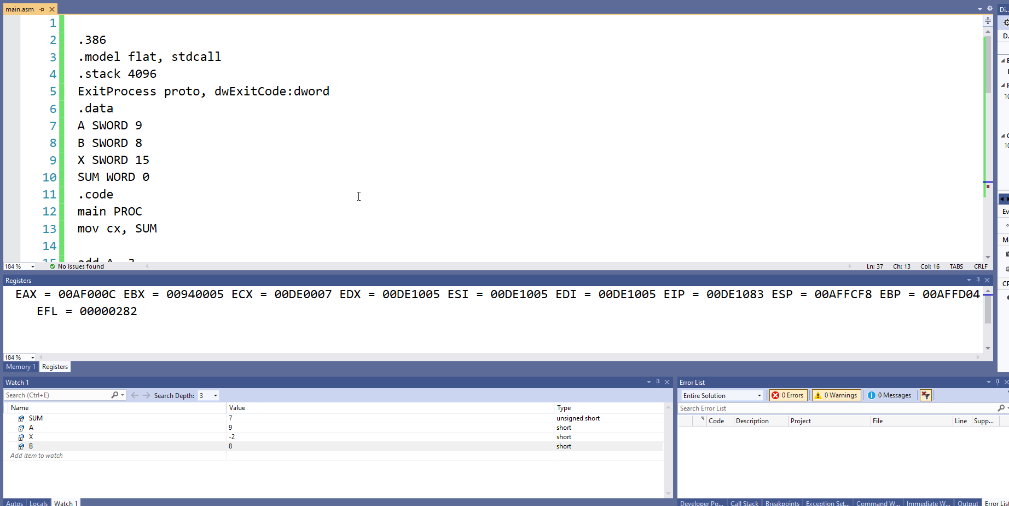
When Part I is evaluated using CPP, the results for each variable are as follows



Thus, for Part 1 with .IF and .ELSE [file titled Gunerli1.asm]



 hakan gunerli  
; csc3210  
; assignment 4 q1   
; this program attempts to short-circuit evaluate the given arguments, with the result sum =7   
  
.386  
.model flat, stdcall  
.stack 4096  
ExitProcess proto,dwExitCode:dword  
  
.data   
SUM WORD 0   
X SWORD 15   
A SWORD 9   
B SWORD 8  
  
.code  
main proc  
MOV CX, 3   
  
ADD A,3  
MOV AX,A  
SUB A,3   
  
SUB B,3   
MOV BX,B   
ADD B,3  
  
.WHILE X>=0  
  
.IF (X != CX) && (X <= AX)  || (X< BX)  
      
    SUB X,2   
    ADD SUM,1    
.ELSE   
    SUB X,1   
.ENDIF   
.ENDW  
    invoke ExitProcess, 0  
  
main endp  
end main

However, there’s also an alternative way to solve this question, which is titled Gunerli1Alt.asm in case the usage of if and else are not allowed, 

; hakan gunerli  
; csc3210  
; assignment 4 q1   
; this program attempts to short-circuit evaluate the given arguments, with the result sum =7   
.386  
.model flat, stdcall  
.stack 4096  
ExitProcess proto, dwExitCode:dword  
.data  
A SWORD 9  
B SWORD 8  
X SWORD 15  
SUM WORD 0  
.code  
main PROC  
mov cx, SUM  
  
add A, 3  
mov ax, A  
sub A,3   
  
sub B, 3  
mov bx, B  
add B,3  
  
beginwhile:  
    cmp X, 0  
    jl endwhile  
    cmp X, 3  
    je L2  
    cmp X, ax  
    jg L2  
    sub X, 2  
    inc SUM  
    jmp beginwhile  
L2:  
    sub X, 1  
    jmp beginwhile  
endwhile:  
    mov cx, SUM  
invoke ExitProcess,0  
main endp  
end main

**PART 2**

; Hakan Gunerli   
; CSC3210  
; ASSIGNMENT 4   
; THIS PROGRAM ATTEMPS TO ANSWER QUESTION 2 WHICH IS TO TEST THE MOST SIG BIT IN REGISTER AL   
  
.386  
.model flat, stdcall  
.stack 4096  
ExitProcess proto, dwExitCode:dword  
.data  
.code  
main PROC  
mov eax, 00000000  
mov ebx, 00000000  
mov ecx, 00000000  
mov edx, 00000000  
  
mov al, 88h ; SIGNED 88   
mov bl, 10000000b ; MASK   
test al, bl  
jnz L1 ; TEST AND JUMP   
shl al, 2  
L1:   
    shr al, 3   
invoke ExitProcess,0  
main endp  
end main