Title: lab8_group48
Name / student ID: 曾千芸109504501, 賈子悅110502514
Group: 48

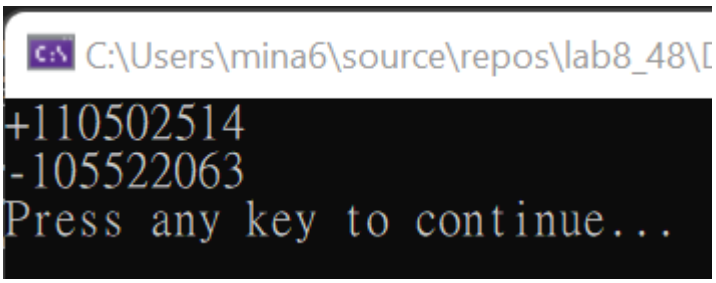
- Screenshots of result and code + explanations

```
INCLUDE Irvine32.inc

FindLargest PROTO,
    ptrarr:PTR SDWORD,
    szarr:DWORD
.data
Ex1Array sdword 105522063,109504501, 110502514
Ex2Array sdword -105522063, -109504501, -110502514
.code

main PROC
    invoke FindLargest,OFFSET EX1Array,LENGTHOF EX1Array      ; displays FindLargest, points to the EX1Array, the length of EX1Array
    invoke FindLargest,OFFSET EX2Array,LENGTHOF EX2Array      ; displays FindLargest, points to the EX2Array, the length of EX2Array
    call WaitMsg                                              ; "Press any key........."
    invoke ExitProcess,0                                      ; end the program
    exit                                                      ; exit
main ENDP
FindLargest PROC,
    ptrarr:PTR SDWORD,                                        ; pointer to array
    szarr:DWORD                                               ; size of the array

    push esi                                                  ; save esi
    push ecx                                                  ; save ecx
    mov eax,80000000h                                         ; store 80000000h into eax
    mov esi,ptrarr                                            ; store pointer to array into esi
    mov ecx,szarr                                             ; store size of array into ecx
L1:
    cmp [esi],eax                                             ; compare each integer with eax
    jl L2                                                     ; jump to L2 if [esi] is less than eax
    mov eax,[esi]                                             ; if [esi] is greater or equal than eax, then store [esi] into eax
L2:
    add esi,4                                                 ; point to the next integer(DWORD is four bytes)
    LOOP L1                                                   ; loop L1
    call WriteInt                                             ; displays: eax
    call Crlf                                                 ; go to the next line
    pop ecx                                                   ; restore ecx
    pop esi                                                   ; restore esi
    ret                                                       ; return
FindLargest ENDP
END main
```



main PROC explanation:
Store EX1Array and EX2Array in FindLargest
call WaitMsg to prevent the code deadlock
ExitProcess, 0 to end the program

FindLargest explanation:
Declare its prototype
Push esi and ecx into stack by order
Store eax in 80000000h

L1 explanation:
Compare [esi] with eax, if the value of [esi] is greater than eax, then store the value in [esi] in eax

Else jump to L2

L2 explanation:
esi+4: point to the next integer(DWORD is four bytes)
Continue running loop L1 until ecx equals 0
call WriteInt: Print the result after the loop end
call Crlf: go to next line
End the program

- Thoughts about the lab:

From this class, we learn how to use Procedure in the program, making it easier to call the same program. Wisely using pointers is also important to this program. Moreover, review the use of loop, compare, and jump, which seems difficult in the last few classes. Furthermore, we truly realized that procedure execute slower than macros because everytime a procedure is called, it is necessary to integrate and link it with the calling program and this takes time. After today's class, we can use them more fluently than before. We both improve our assembly language abilities in this class, and have a more clear view to programming.