Assembly Language Report (Week 4)

Group 7

104502517戴逸任

104502518劉冠聲

Program CODE:

.data

myID BYTE "104502518"

size\_ID = LENGTHOF myID

myID2 BYTE "104502517"

size\_ID2 = LENGTHOF myID2

.code

Convert PROC USES eax ;the beginning of convert function and push eax into the stack

L1: ;loop

mov al,[esi] ;al=the value of esi

add al,11h ;al=al+11h

mov [esi],al ;the value of esi=al

inc esi ;esi=esi+1

loop L1 ;loop

ret ;convert function and return to end

Convert ENDP ;convert function end

Convert2 PROC ;the beginning of convert2 function

PUSH eax ;push eax inti the stack

L1: ;loop

mov al,[esi] ;al=the value of esi

add al,11h ;al=al+11h

mov [esi],al ;the value of esi=al

inc esi ;esi=esi+1

loop L1 ;loop

POP eax ;pop the eax out from stack

ret ;convert2 function and return to end

Convert2 ENDP ;convert2 function end

start@0 PROC

mov eax,9999h ;eax=9999h

mov ebx,9999h ;ebx=9999h

mov edx,9999h ;edx=9999h

mov esi, OFFSET myID ;esi=the address of myID

mov ecx, size\_ID ;ecx=9

call Convert ;run convert function

mov esi, OFFSET myID2 ;esi=the address of myID2

mov ecx, size\_ID2 ;ecx=9

call Convert2 ;run convert2 function

exit

start@0 ENDP

END start@0

Program Step & Register state:

mov eax,9999h ;eax=9999h

mov ebx,9999h ;ebx=9999h

mov edx,9999h ;edx=9999h

mov esi, OFFSET myID ;esi=0x00404000

mov ecx, size\_ID ;ecx=9

call Convert

mov esi, OFFSET myID2 ;esi=0x00404009 eax=9999h

mov ecx, size\_ID2 ;ecx=9

call Convert2 ;eax=9999h

L1:

mov al,[esi] ;al=0x31🡪al=0x30🡪……

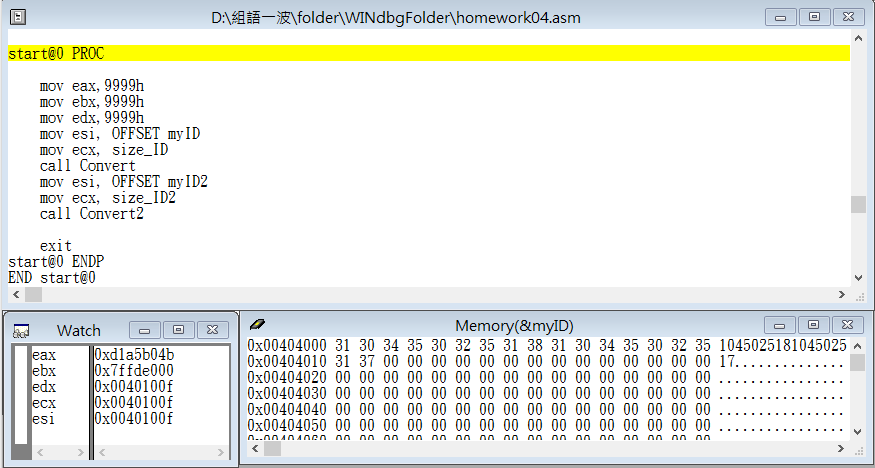
add al,11h ;al=0x42🡪al=0x41🡪……

mov [esi],al ;al=0x42🡪al=0x41🡪……

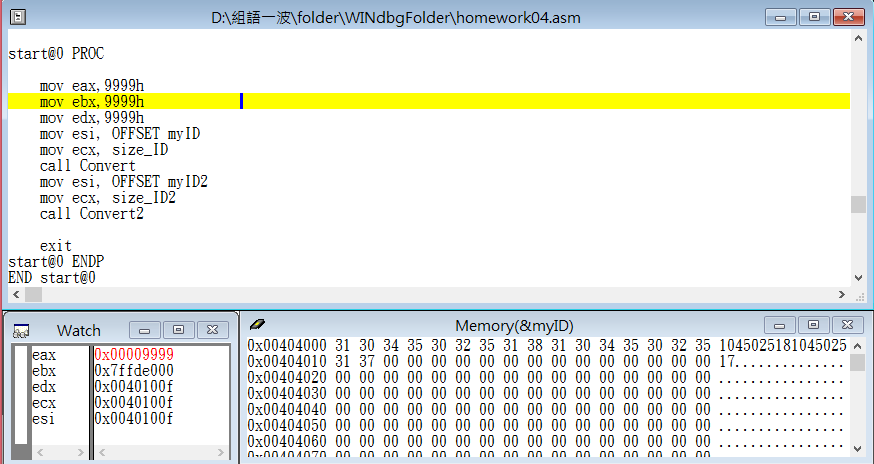
inc esi ;esi=0x00404001🡪esi=0x00404002🡪……

loop L1 ;loop

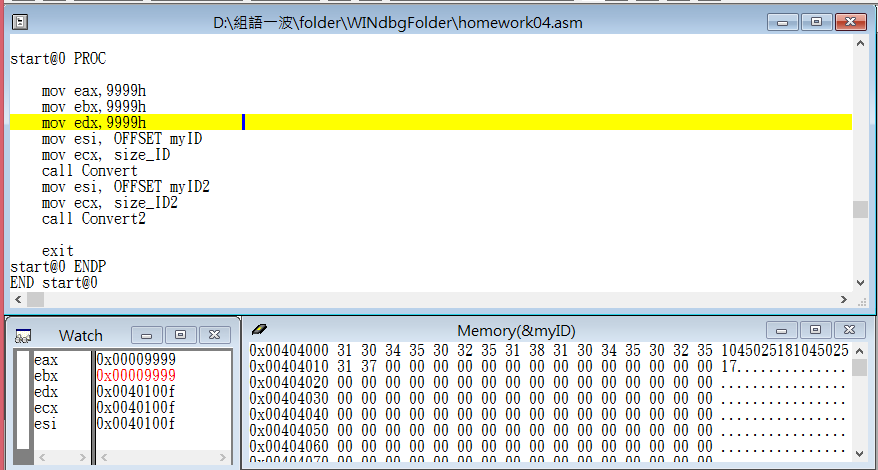
Picture & Discription:



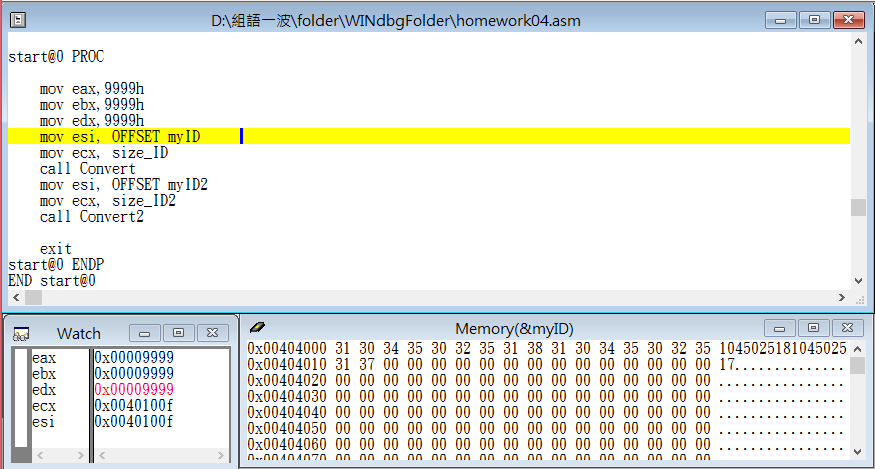
Step1: Start the program



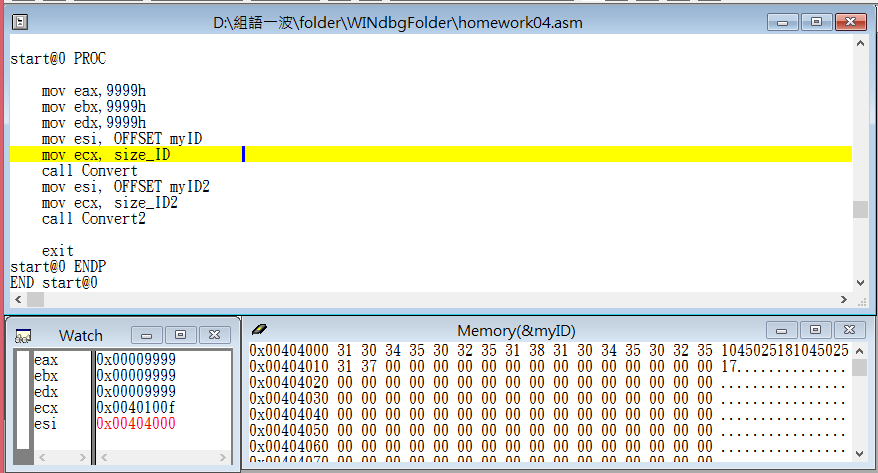
Step2: store 9999h into register eax



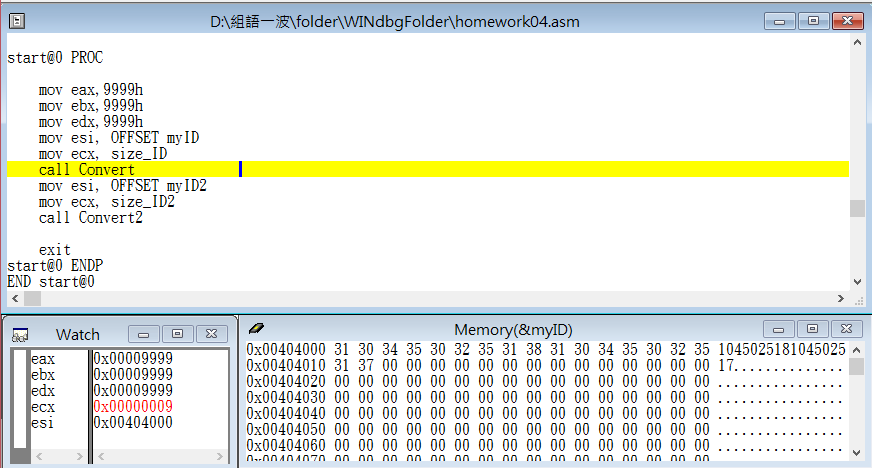
Step3: store 9999h into register ebx



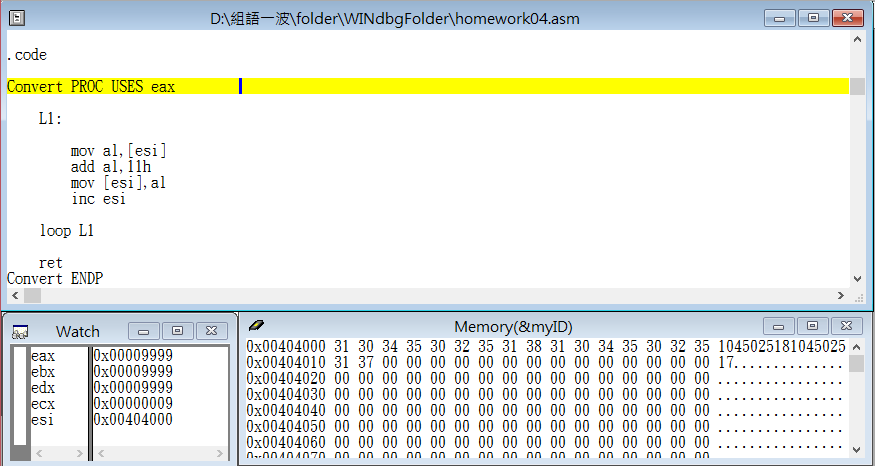
Step4: store 9999h into register edx



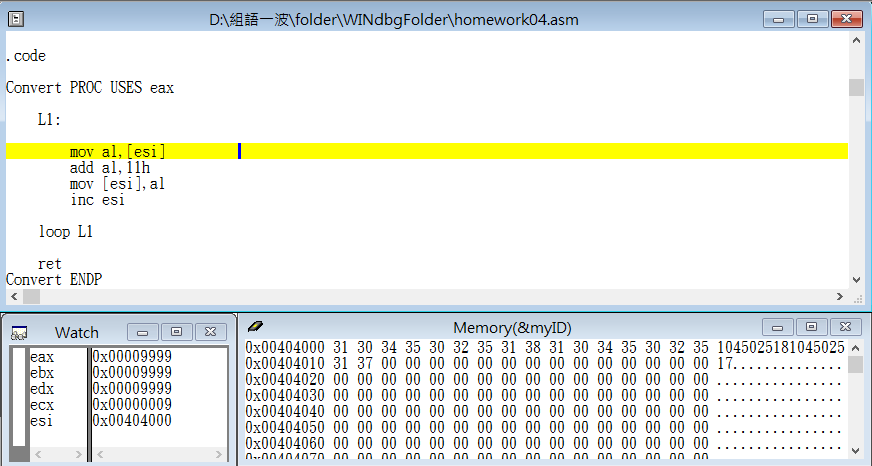
Step5: let esi point to the variable myID address



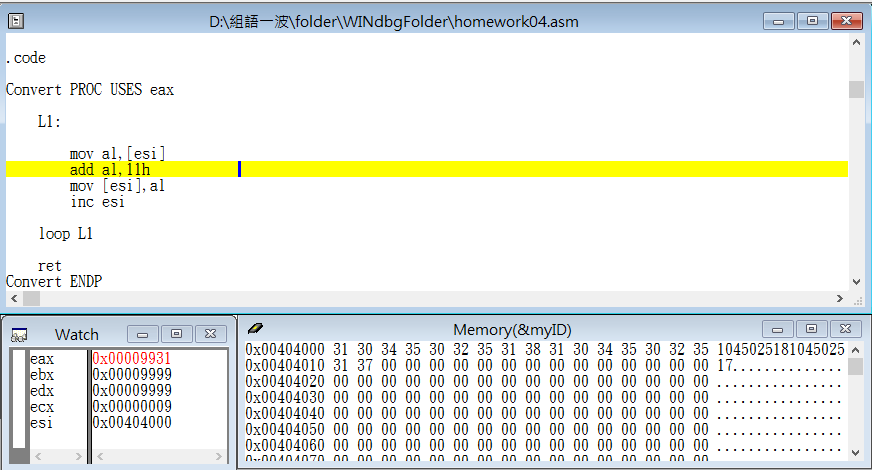
Step6: use myID length to be the loop counter



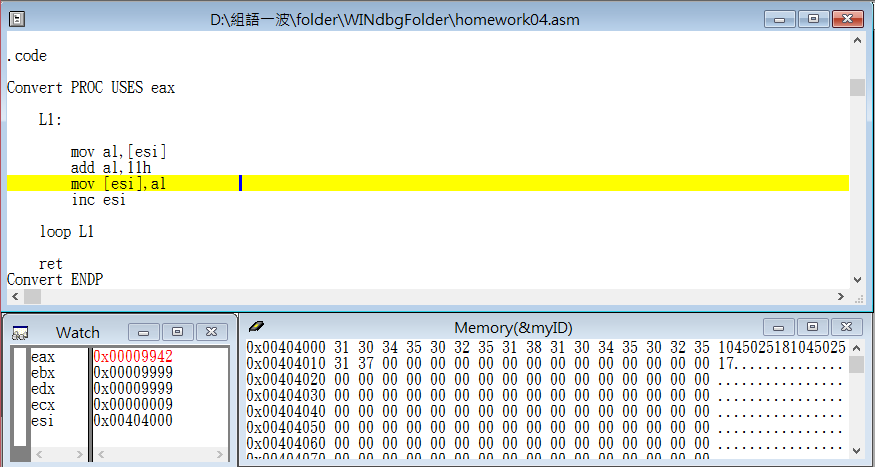
Step7: call the convert function and uses eax (push eax into the stack)



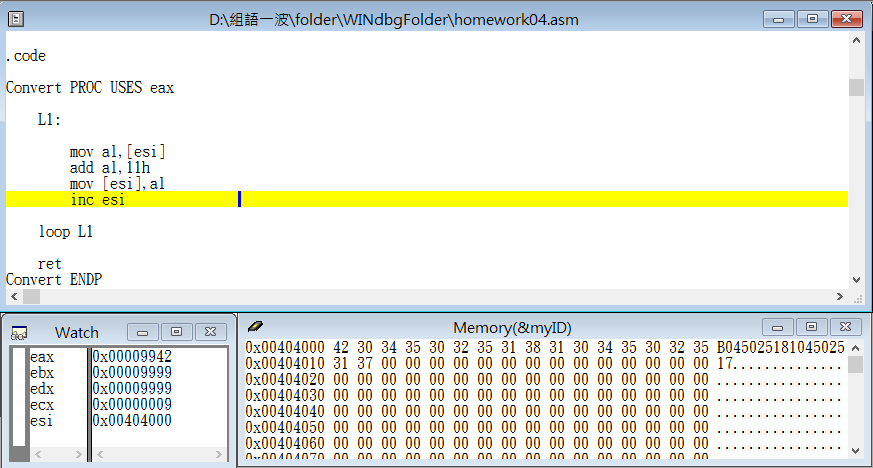
Step8: start L1



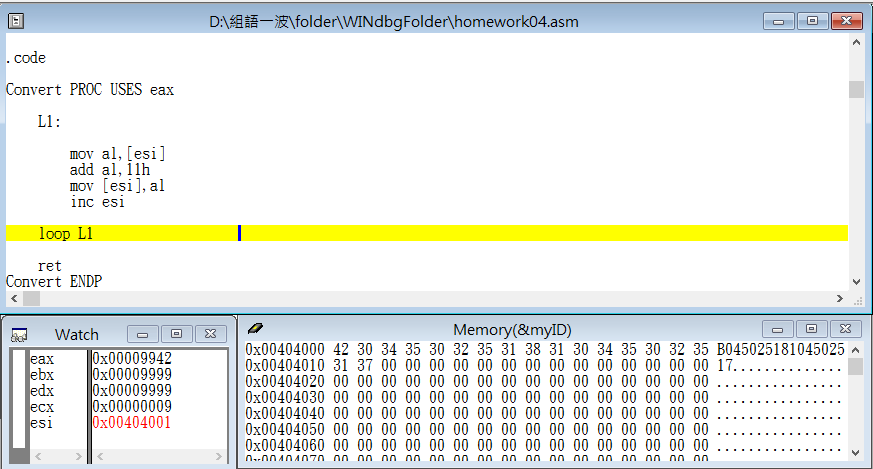
Step9: store the value that esi point at into the register al



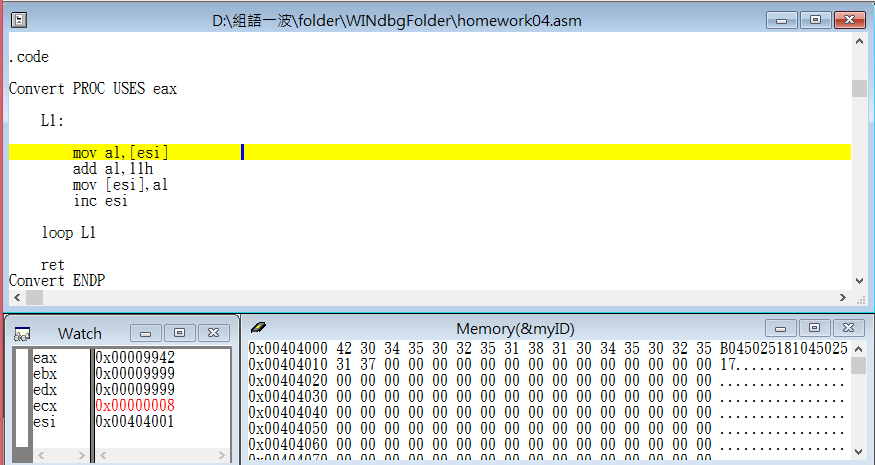
Step10: add 11h to al



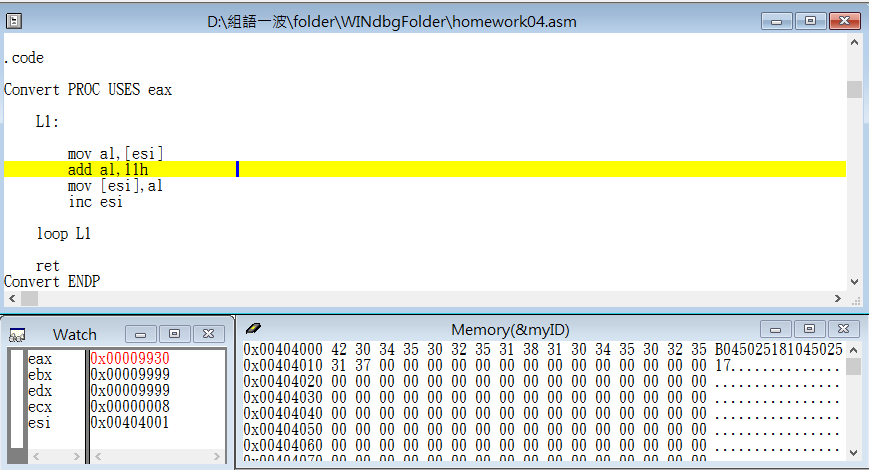
Step11: store the value in al into the address that esi point at(1 -> B)



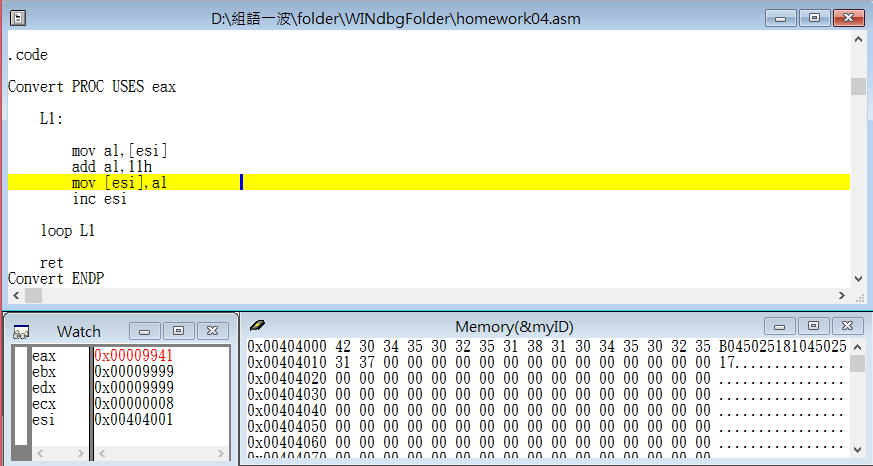
Step12: move esi to point next address



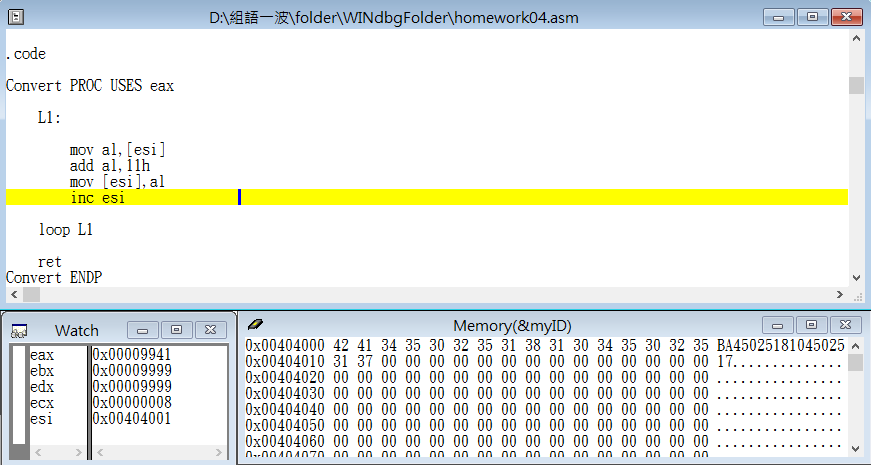
Step13: loop L1 (totally loop 9 times)



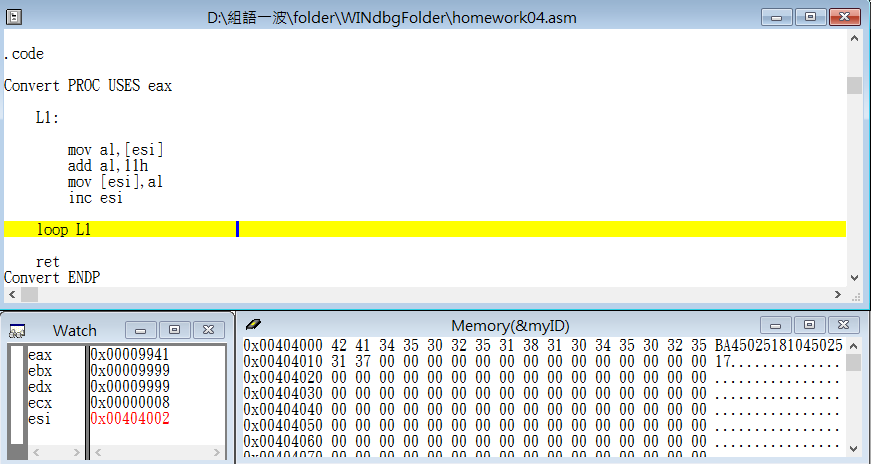
Step14: store the value that esi point at into the register al



Step15: add 11h to al



Step16: store the value in al into the address that esi point at(0 -> A)

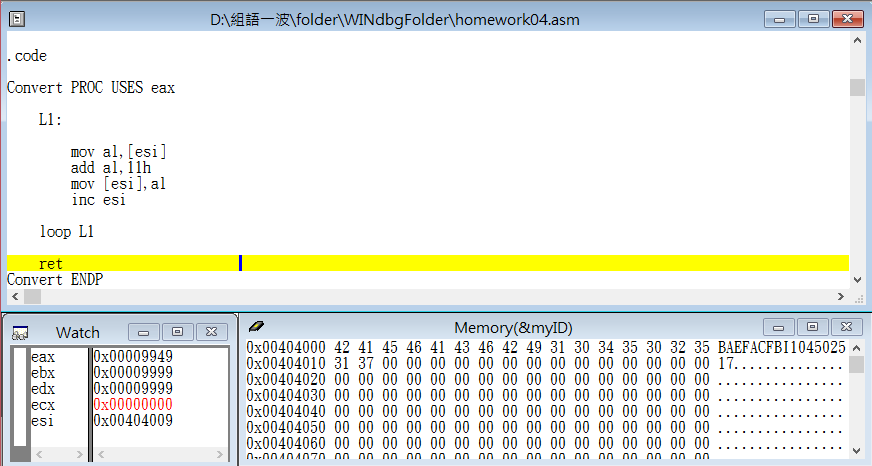


Step17: move esi to point next address

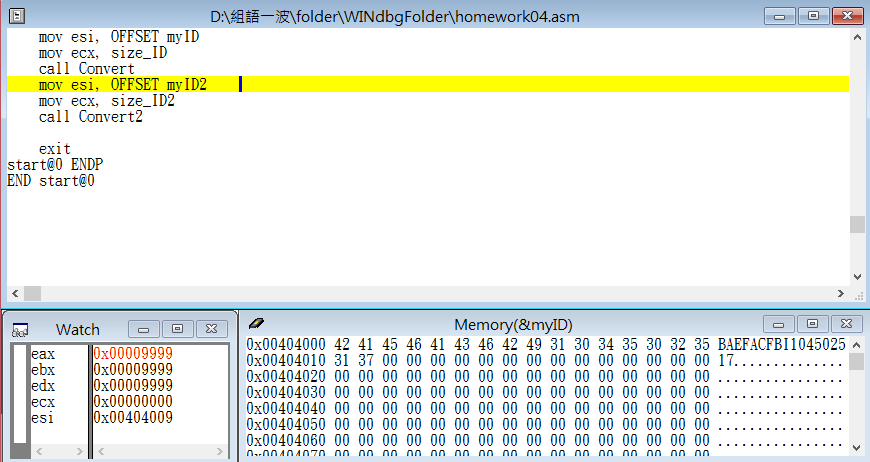
.

.

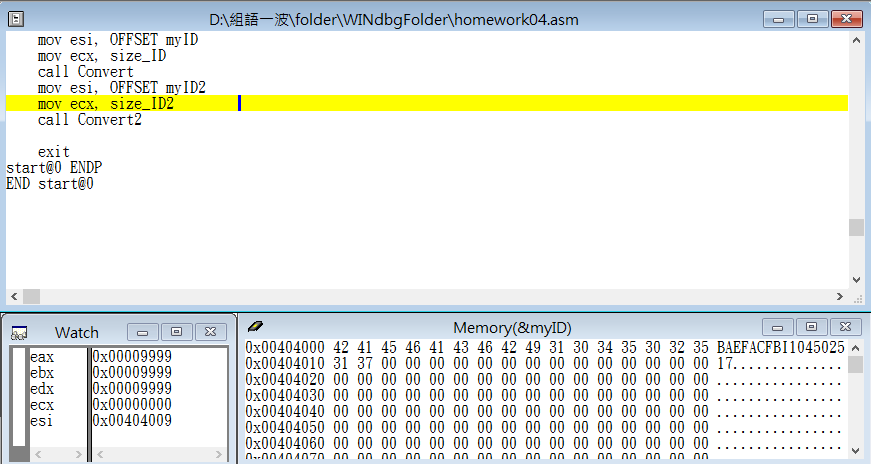
.



Step18: loop L1 end and return to main

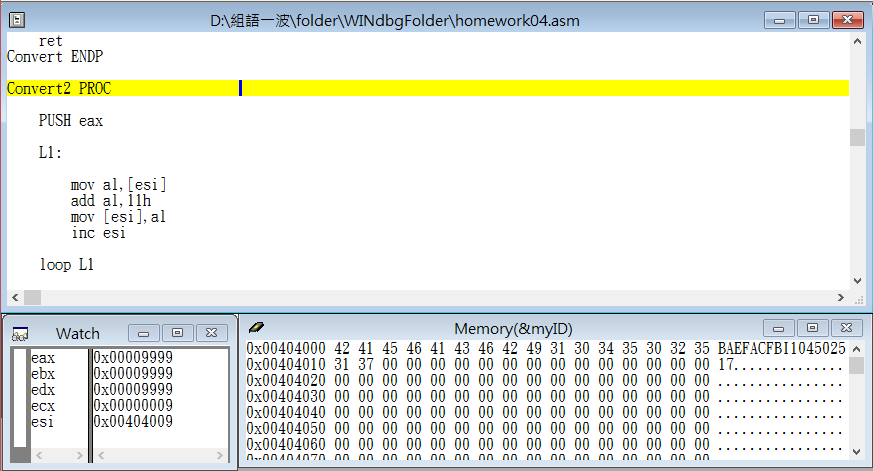
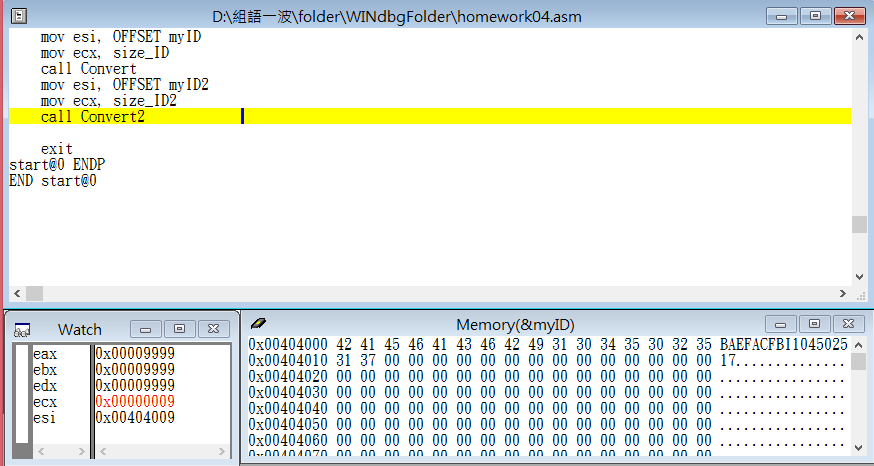


Step19: move esi to point n



Step20: let esi point to the variable myID2 address

Step21: use myID2 length to be the loop counter

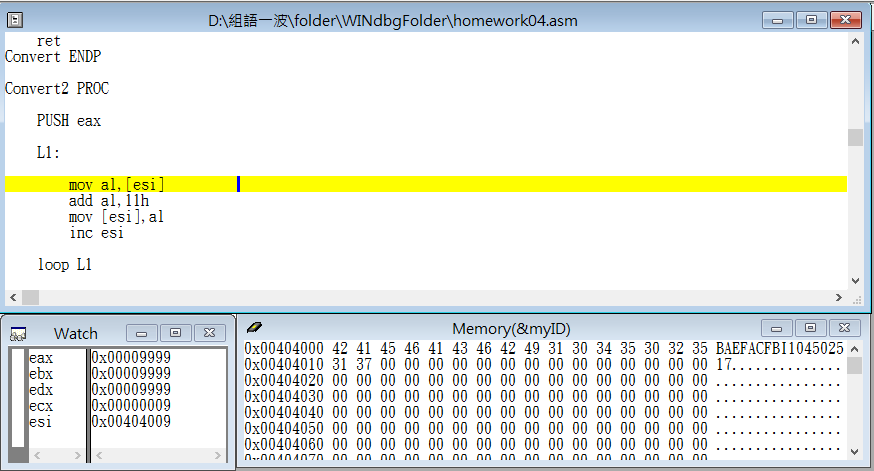


Step22: start Convert2 function

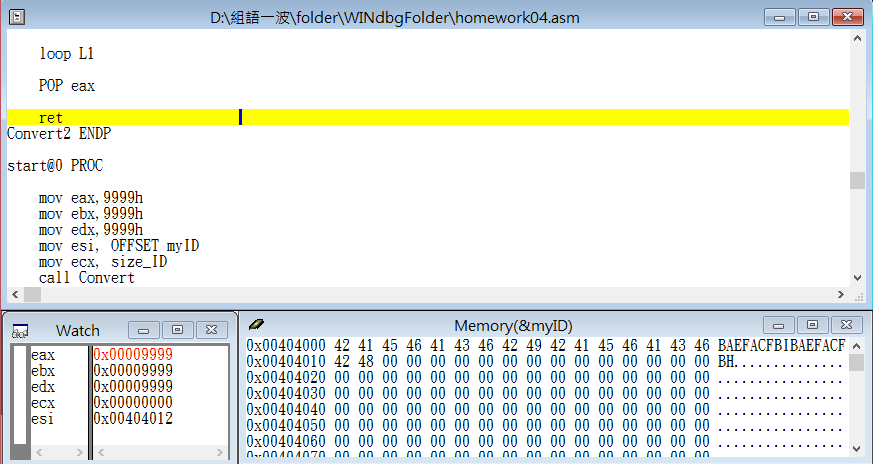
.

.

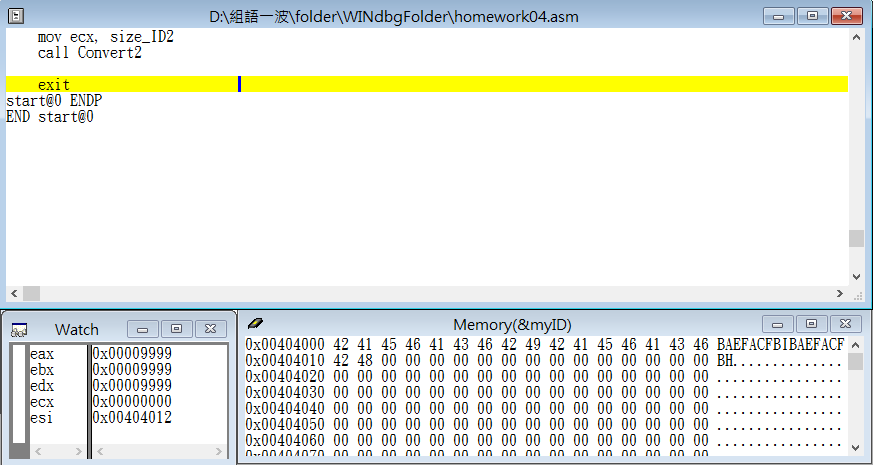
.



Step23: push eax into the stack



Step24: pop the eax from stack then return to main



Step25: the end

Review:

The exercise this week is to store our student ID which is in form of string into a memory, and use stack to finish the exercise. In the same time, we have use two different program to achieve the same consequence, then the first method is use “USES” the help us finish it, and the other method is that we have to use “POP” and “PUSH” to finish it.

Another problem we met this time is we are not clearly know how to precisely transform the number into correct letter with ASCII code, we had a mess of ASCII code transformation between decimal and hexadecimal, finally, we transformed the student number into the right letter, and we think it is a good exercise for us to learn assembly language.