Assembly Language Report (HW2)

104502518劉冠聲

Program CODE:

.data

BitStrs BYTE 8 dup(?) ;initialize BitStrs

ChStrs BYTE "########" ;initialize ChStrs

BYTE "########"

BYTE "## ##"

BYTE "########"

BYTE "########"

BYTE "## ##"

BYTE "########"

BYTE "########"

.code

change PROC

L2:

mov ah,[esi] ;use ah to store value of ChStrs

cmp ah,32 ;compare ChStrs is # or blank, if it is # jump to L3

jnz L3

mov al,48 ;it is blank, so store '0' ASCII code into al

jmp L4

L3:

mov al,49 ;it is #, so store '1' ASCII code into al

L4:

mov [ebx],al ;store ASCII into BitStrs

inc ebx ;point to next BitStrs

inc esi ;point to next ChStrs

inc dl ;increase counter dl

cmp dl,8 ;if dl is not 8,jump to L2 to continue the same line

jnz L2

ret ;return to main

change ENDP

start@0 PROC

mov ecx,8 ;count is 8

mov esi,OFFSET ChStrs ;point to ChStrs

L1:

mov ebx,OFFSET BitStrs ;use ebx to point BitStrs

mov dl,0 ;initialize counter dl

call change ;call the function change

mov dl,0 ;initialize counter dl to be used to output BitStrs

mov ebx,OFFSET BitStrs ;point to the beginning of BitStrs

L5: ;L5 for output BitStrs

mov al,[ebx]

call WriteChar

inc ebx

inc dl

cmp dl,8

jnz L5

call Crlf ;jump to next line in terminal

loop L1 ;loop L1 for 8 times

call WaitMsg ;call the function WaitMsg

exit ;end

start@0 ENDP

END start@0

Program Step & Register state:

.code

change PROC

L2:

mov ah,[esi] ah=23🡪23🡪…🡪20🡪20…🡪23🡪23…

cmp ah,32

jnz L3

mov al,48 al=48

jmp L4

L3:

mov al,49 al=49

L4:

mov [ebx],al

inc ebx ebx=0x00405001🡪0x00405002……

inc esi esi=0x00405009🡪0x00405010……

inc dl dl=1🡪2🡪3……

cmp dl,8

jnz L2

ret

change ENDP

start@0 PROC

mov ecx,8 ecx=0x0008

mov esi,OFFSET ChStrs esi=0x00405008

L1:

mov ebx,OFFSET BitStrs ebx=0x00405000

mov dl,0 dl=0x00

call change

mov dl,0 dl=0x00

mov ebx,OFFSET BitStrs ebx=0x00405000

L5:

mov al,[ebx] al=1🡪1🡪……0🡪0🡪……1🡪1🡪……

call WriteChar

inc ebx ebx=0x00405001🡪0x00405002……

inc dl dl=1🡪2🡪3……

cmp dl,8

jnz L5

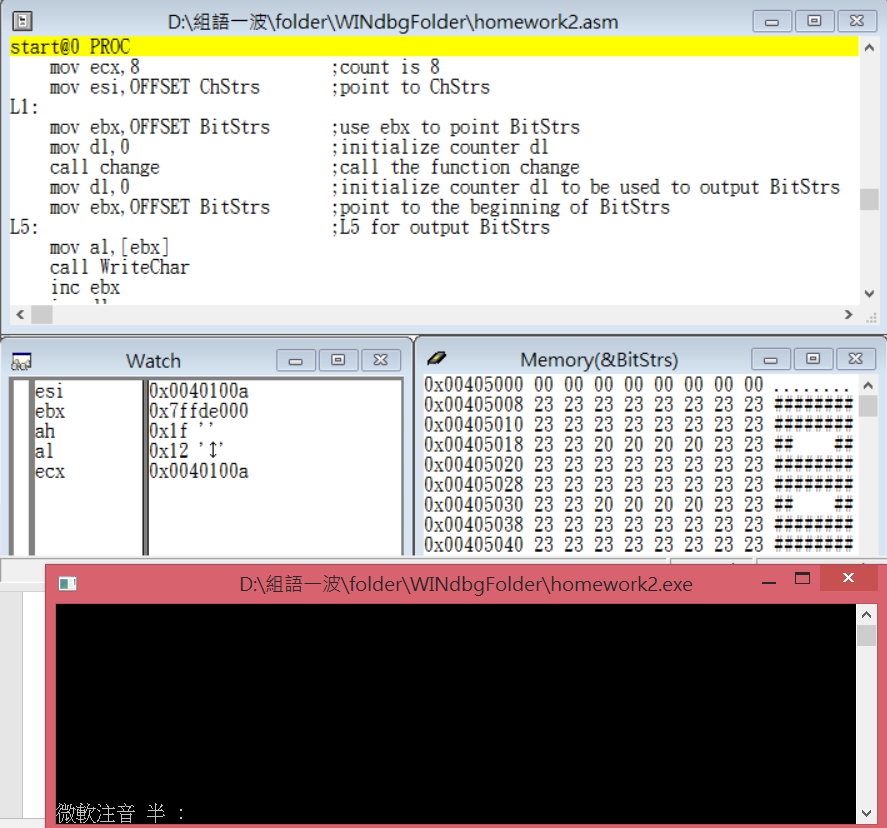
call Crlf

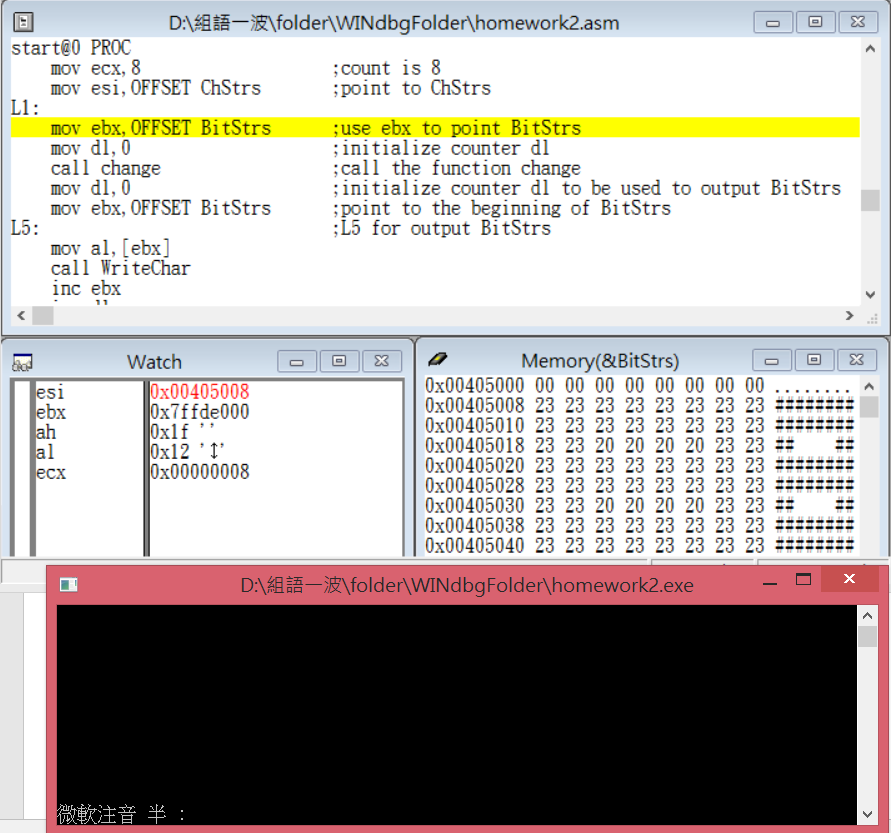
loop L1

call WaitMsg

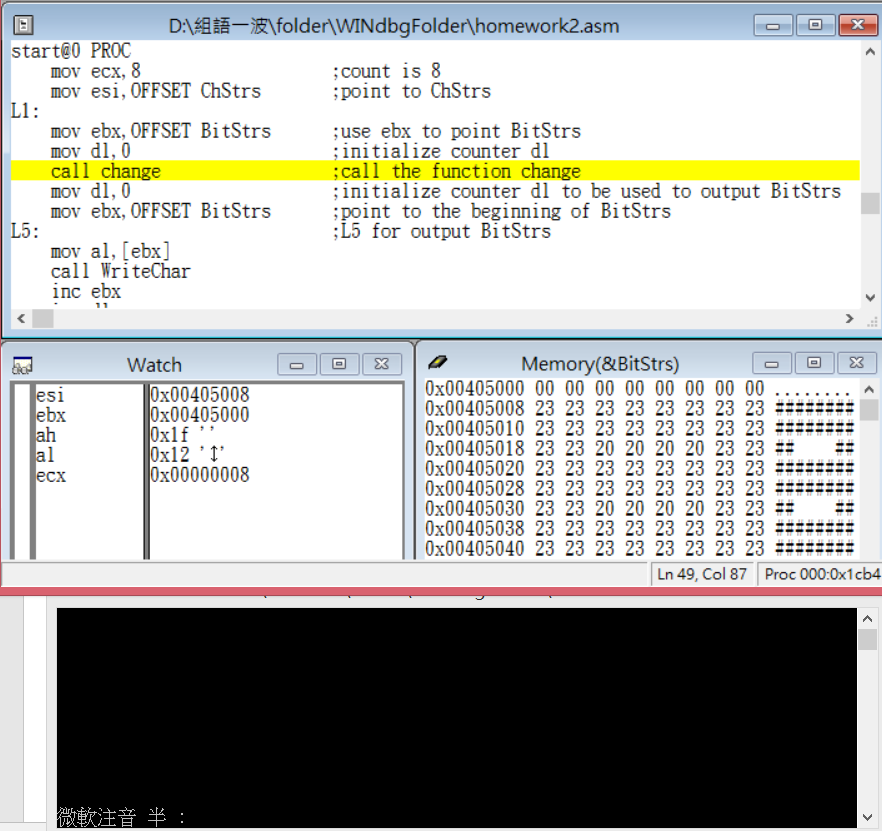
exit

Picture & Discription:

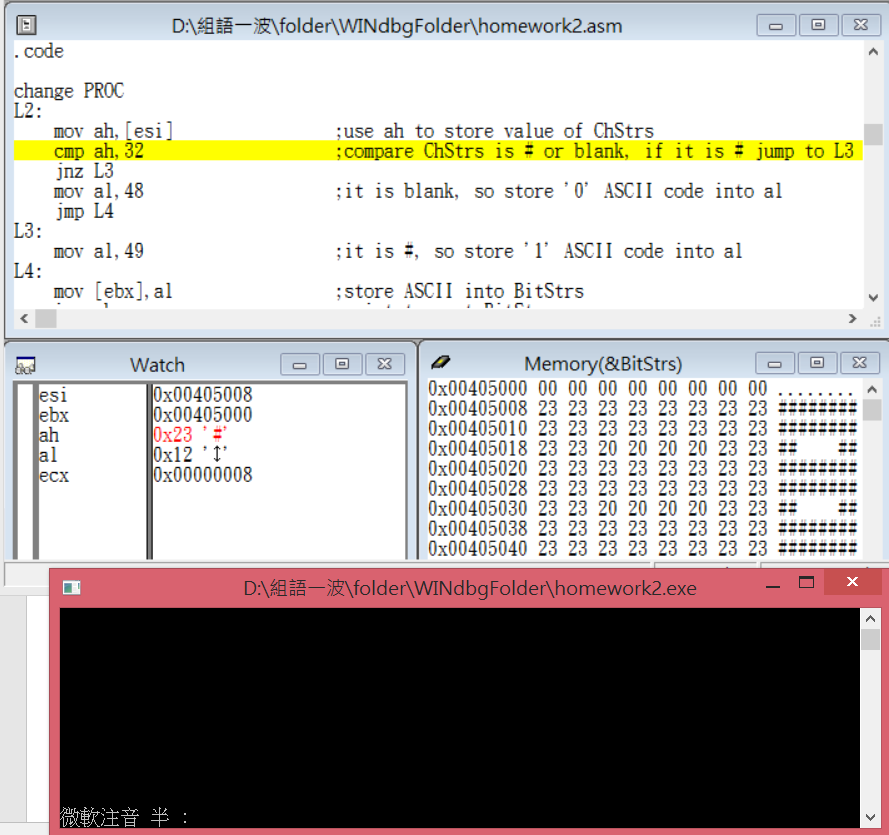
Step1: start the program



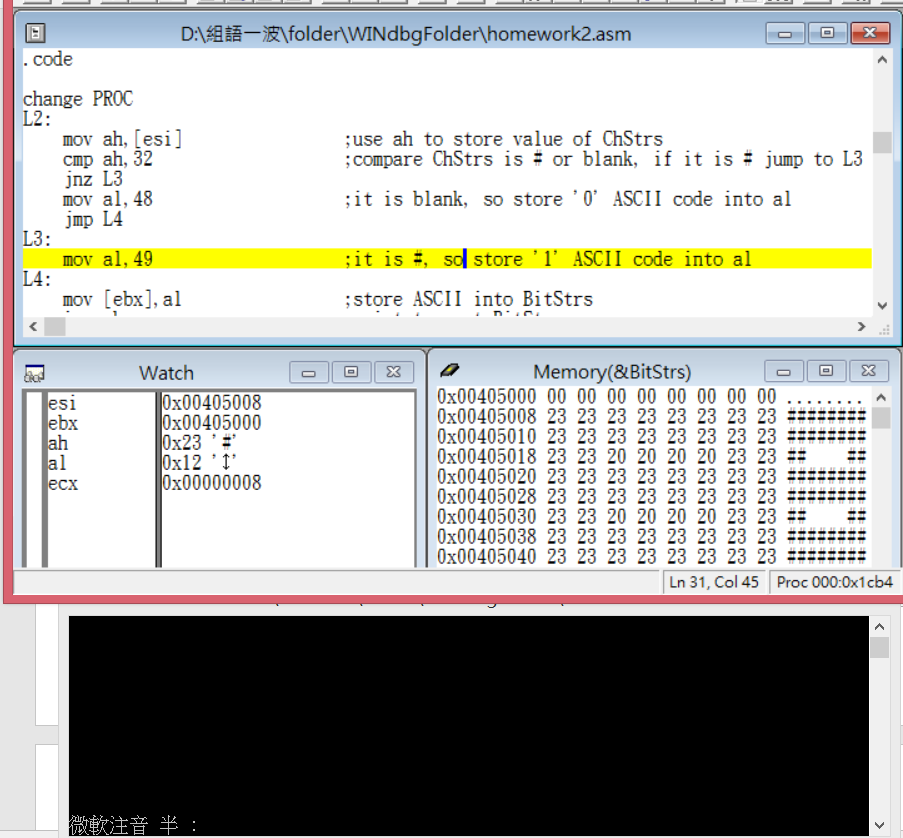
Step2: set pointer and loop times



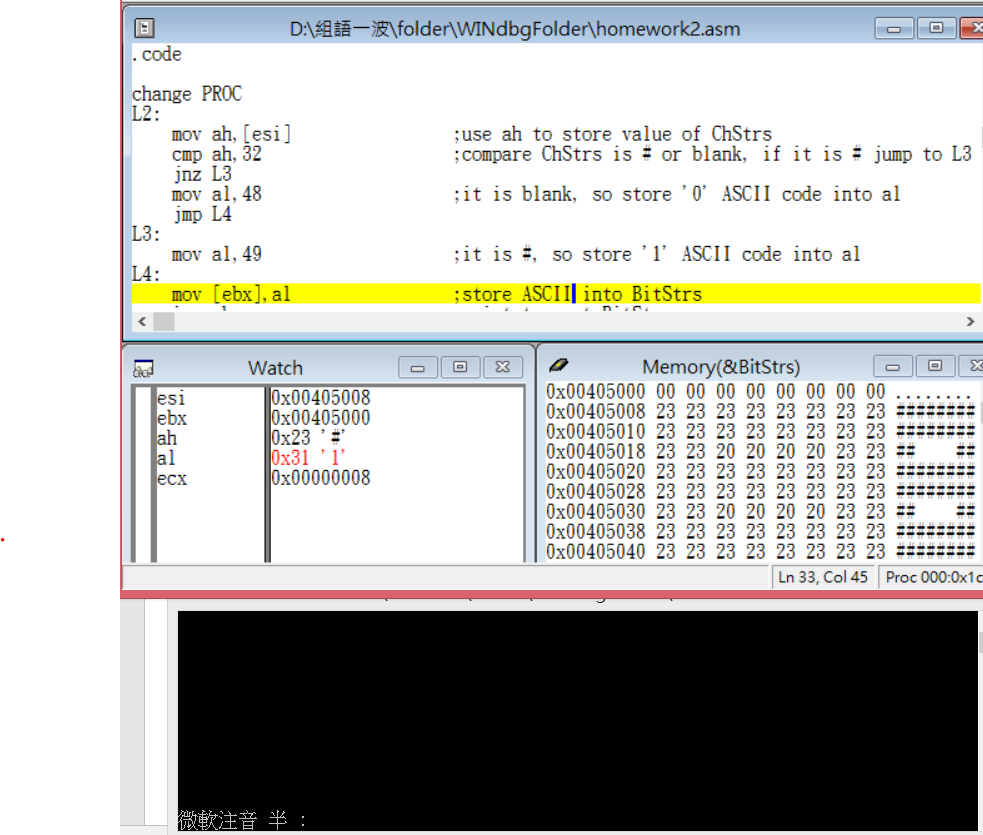
Step3: set some initial in the funtionl and call it



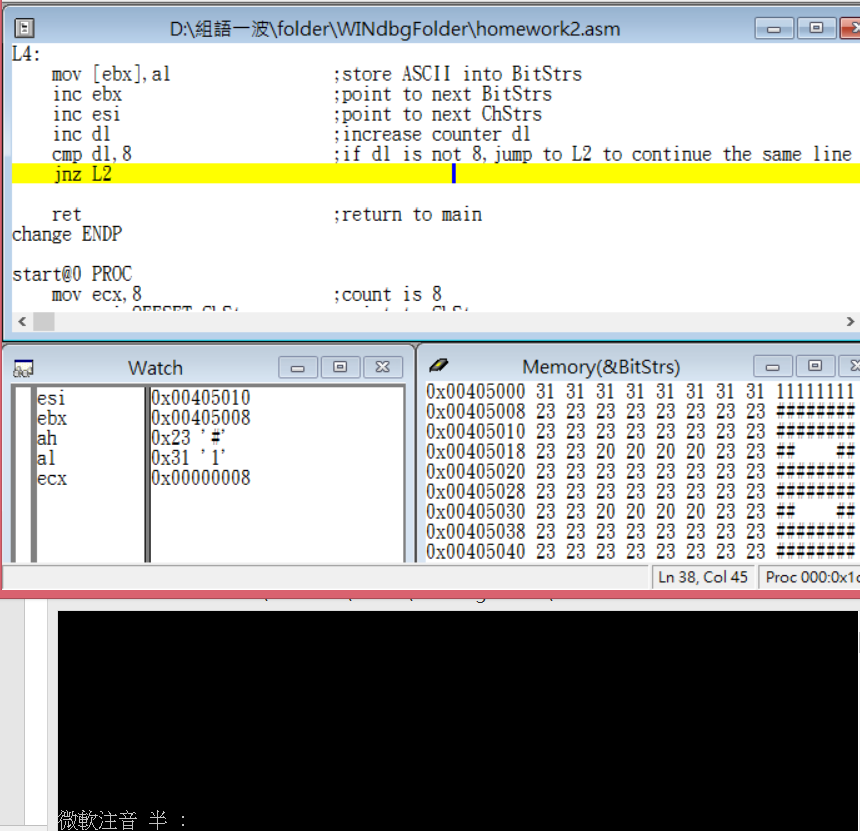
Step4: get value in the pointer and judge it, then jump



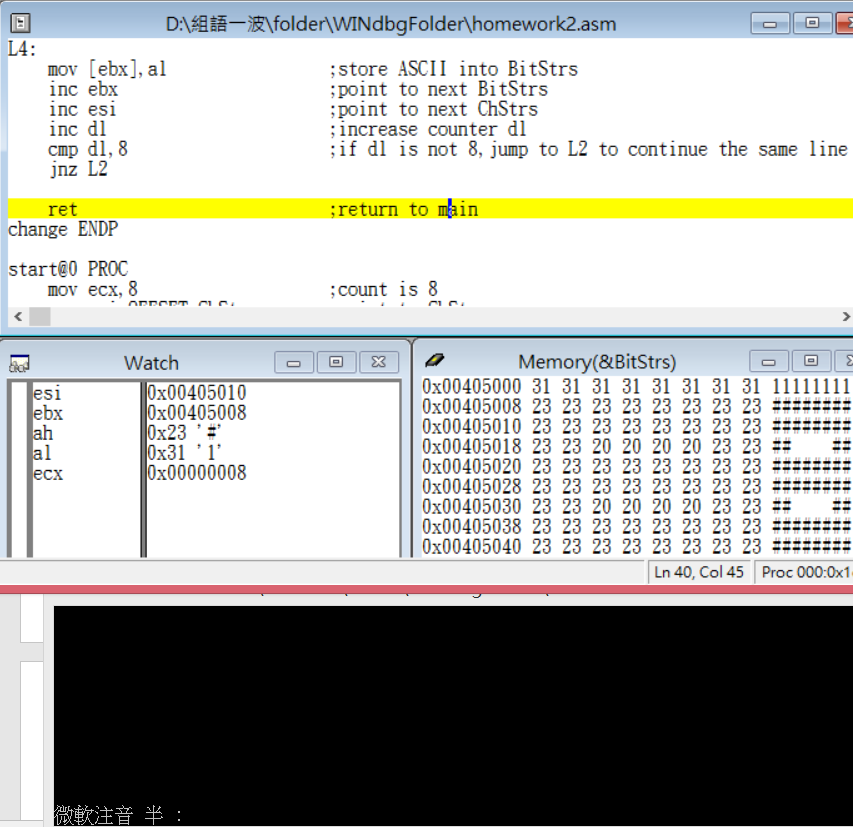
Step5: give it correspond value



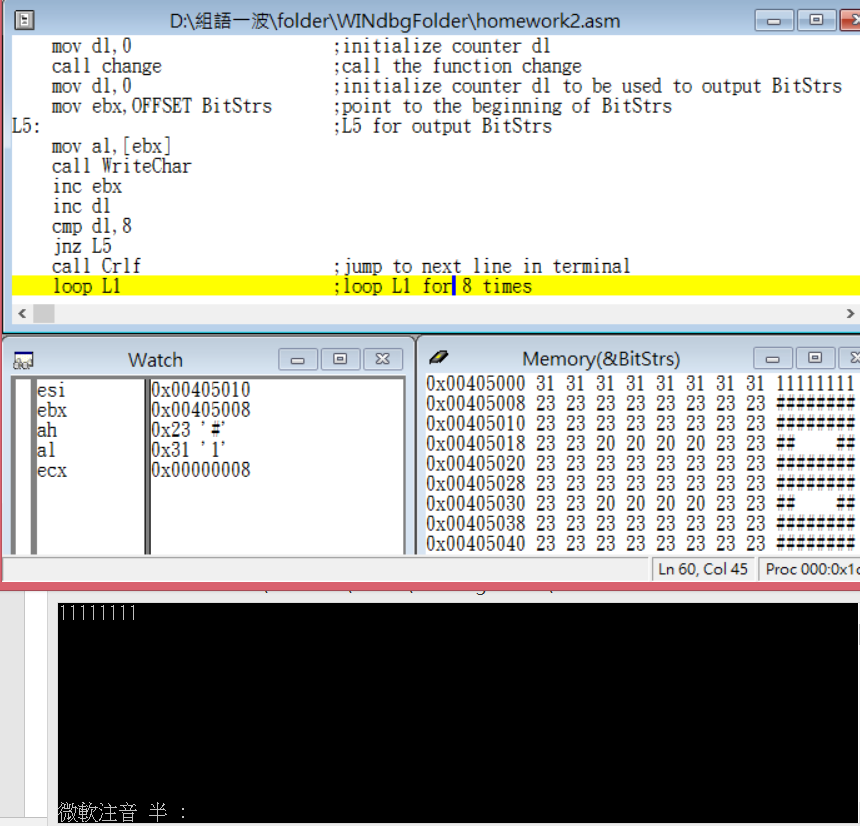
Step6: store the result



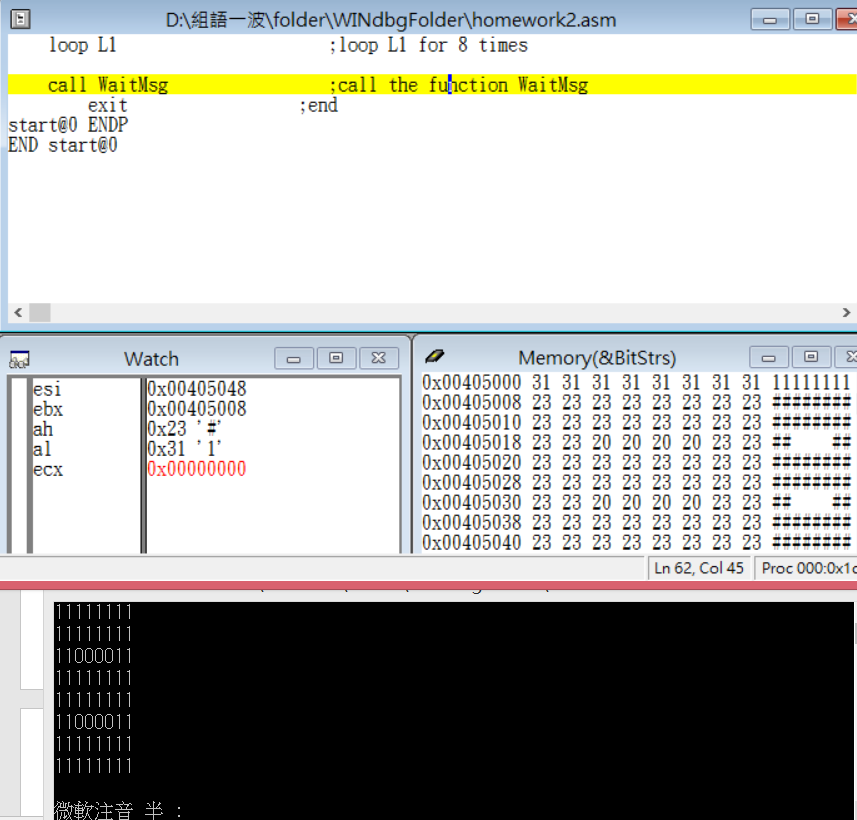
Step7: repeat it 8 times



Step8: return to main function



Step9: output all the result



Step10: repeat all 8 times

Review:

I felt confused when I see this homework in the beginning, because I didn’t know how to call terminal in the assembly, until I found the terminal would come up when I open the execute file in the window debugger soon later.

And then I started to think how to use the loop and jump command to make my last number of sudentID pattern changed from ‘#’ and ‘ ‘ to ‘1’ and ‘0’, and output in the terminal.

When I finished all code, I just thought out I can use USED command to let ecx be used in change function but not cmp and jnz. Well, it came out the same result so I was lazy to revise it.