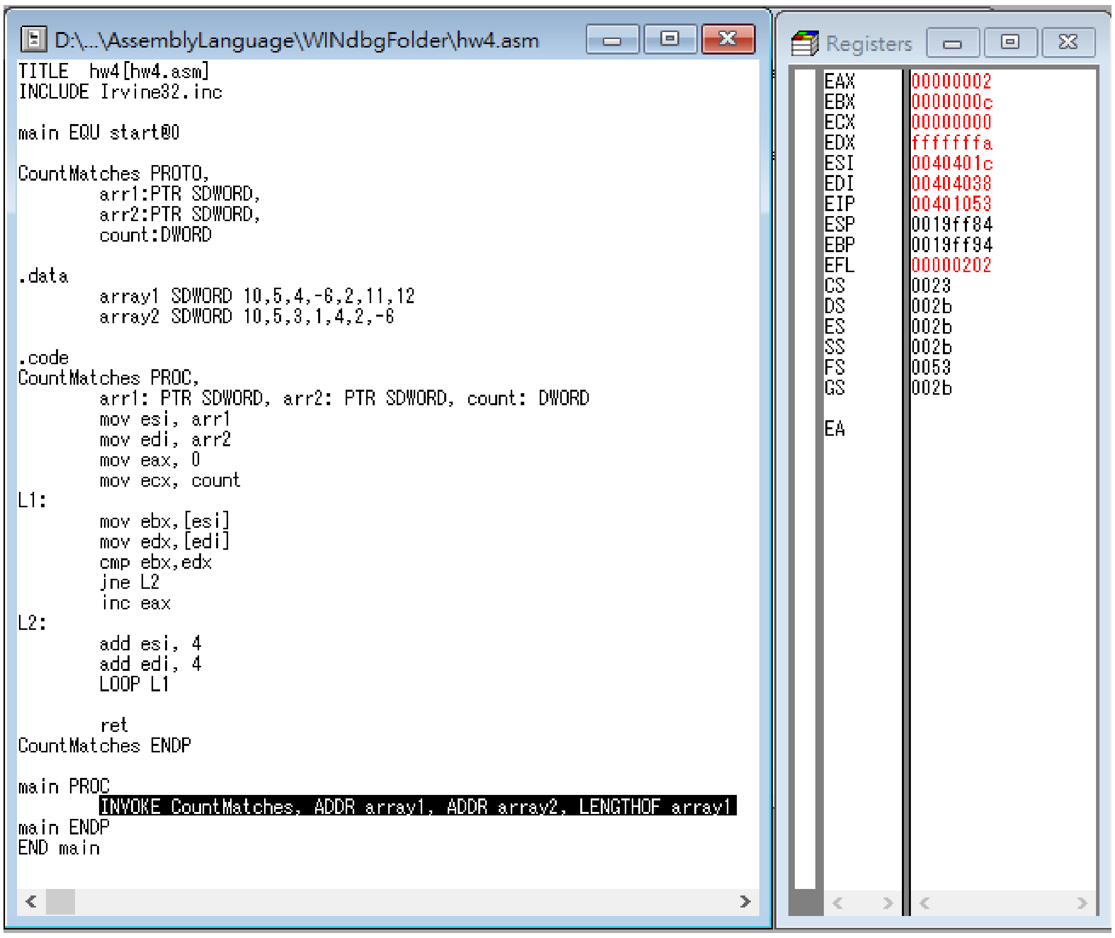
HW#04Counting Matching Elements

資管 3A 102403015 程祥恩

Code – Basic:

```
TITLE  hw4[hw4.asm]
INCLUDE Irvine32.inc

main EQU start@0

CountMatches PROTO,
    arr1:PTR SDWORD,
    arr2:PTR SDWORD,
    count:DWORD

.data
    array1 SDWORD 10,5,4,-6,2,11,12
    array2 SDWORD 10,5,3,1,4,2,-6

.code
CountMatches PROC,
    arr1: PTR SDWORD, arr2: PTR SDWORD, count: DWORD
    mov esi, arr1
    mov edi, arr2
    mov eax, 0
    mov ecx, count
L1:
    mov ebx,[esi]
    mov edx,[edi]
    cmp ebx,edx
    jne L2
    inc eax
L2:
    add esi, 4
    add edi, 4
    LOOP L1

    ret
CountMatches ENDP

main PROC
    INVOKE CountMatches, ADDR array1, ADDR array2, LENGTHOF array1
main ENDP
END main
```

Result – Basic:



D:\...\AssemblyLanguage\WINdbgFolder\hw4.asm

```
TITLE  hw4[hw4.asm]
INCLUDE Irvine32.inc

main EQU start@0

CountMatches PROTO,
        arr1:PTR SDWORD,
        arr2:PTR SDWORD,
        count:DWORD

.data
        array1 SDWORD 10,5,4,-6,2,11,12
        array2 SDWORD 10,5,3,1,4,2,-6

.code
CountMatches PROC,
        arr1: PTR SDWORD, arr2: PTR SDWORD, count: DWORD
        mov esi, arr1
        mov edi, arr2
        mov eax, 0
        mov ecx, count
L1:
        mov ebx,[esi]
        mov edx,[edi]
        cmp ebx,edx
        jne L2
        inc eax
L2:
        add esi, 4
        add edi, 4
        LOOP L1

        ret
CountMatches ENDP

main PROC
        INVOKE CountMatches, ADDR array1, ADDR array2, LENGTHOF array1
main ENDP
END main
```

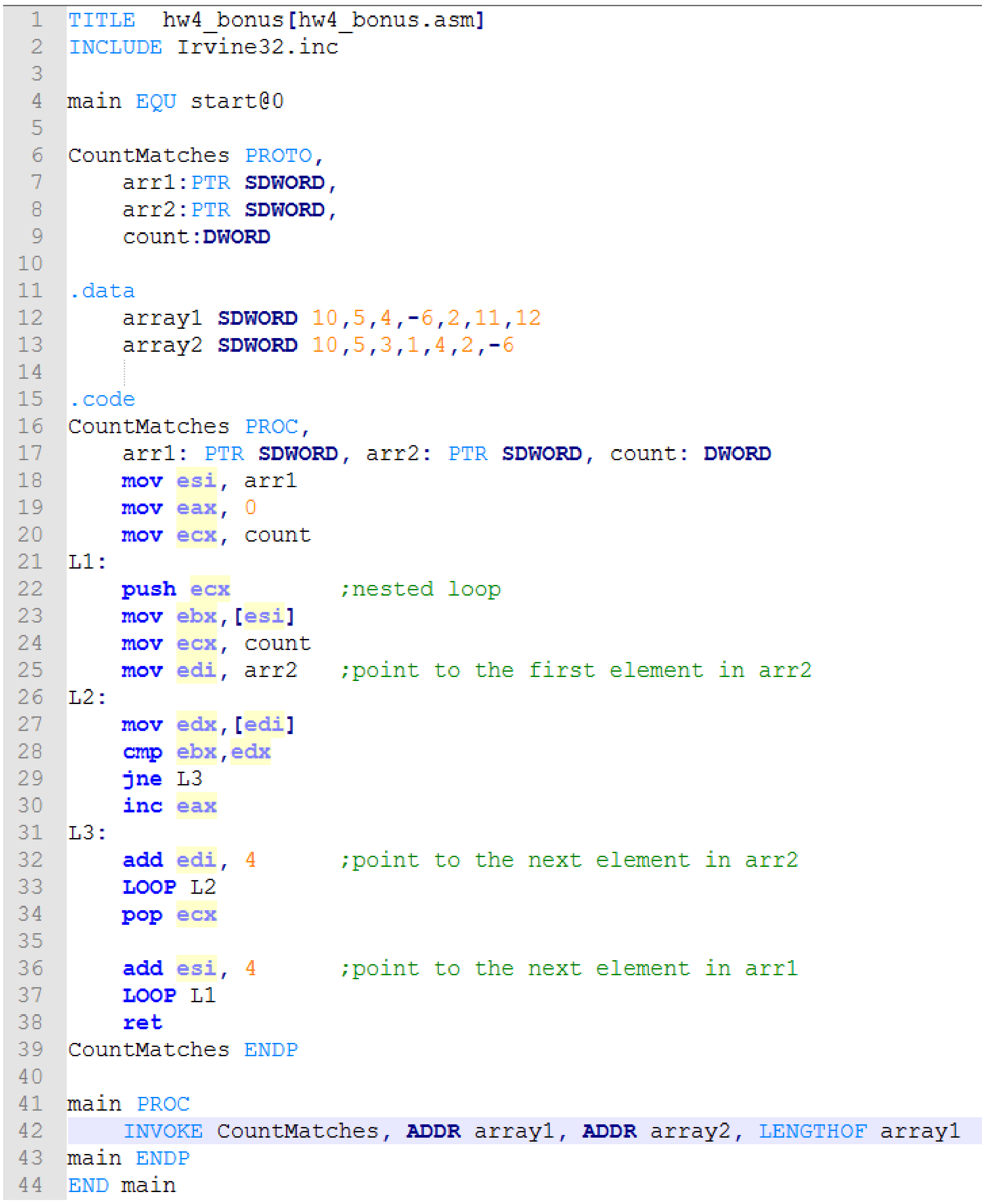
Registers

```
EAX  00000002
EBX  0000000c
ECX  00000000
EDX  fffffffa
ESI  0040401c
EDI  00404038
EIP  00401053
ESP  0019ff84
EBP  0019ff94
EFL  00000202
CS   0023
DS   002b
ES   002b
SS   002b
FS   0053
GS   002b

EA
```

Explanation – Basic:

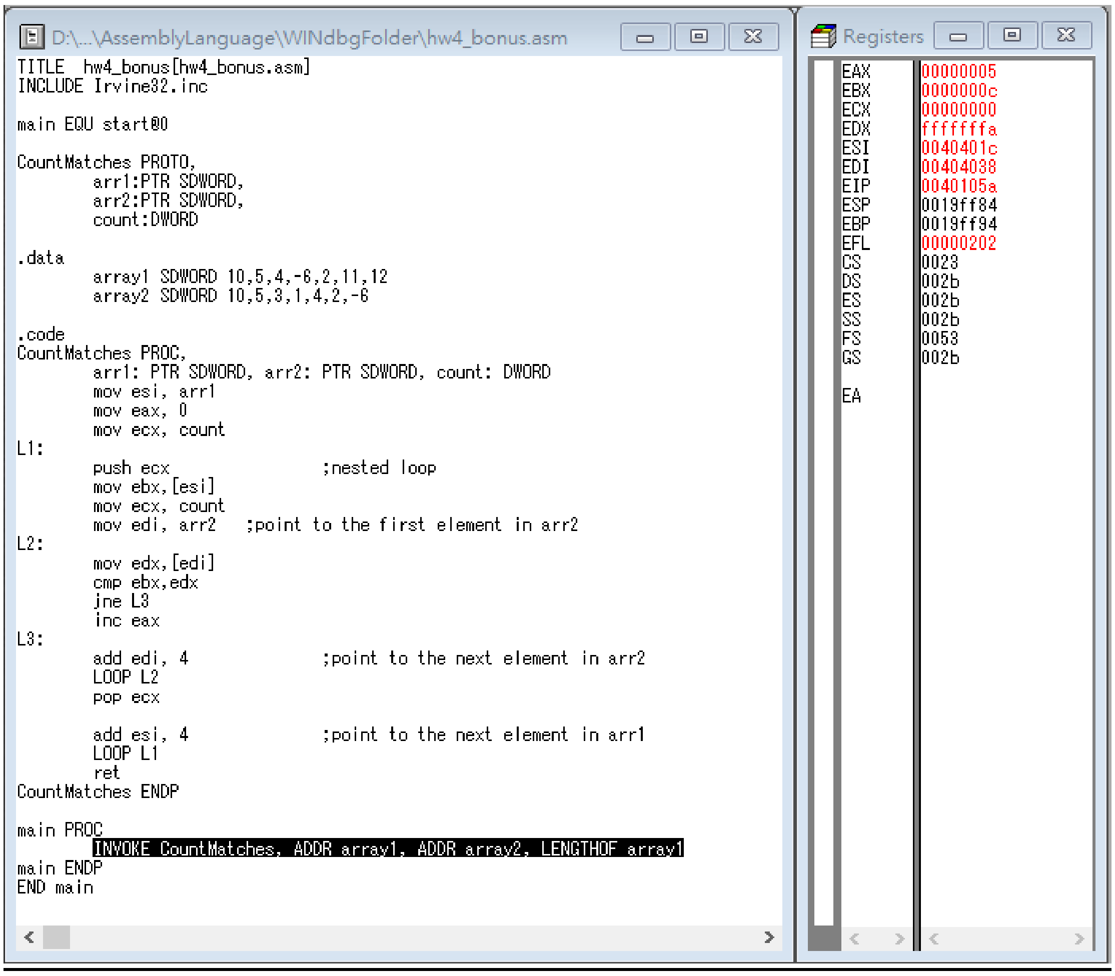
首先要建立出 CountMatches PROC 的原型(PROTO)，原型包含定義程序名稱、傳入參數的型別與數目，如此一來才能用 INVOKE 呼叫 CountMatches PROC。在 CountMatches 中，先把 eax 設為零以便於存放結果，並建立一個迴圈，其重複次數等於陣列的元素數量，接者比對 arr1[i]和 arr2[i]的值，如果值相等，就會執行 inc eax 使 eax 加一，若否則直接跳到 L2，CountMatches PROC 結束後，eax 中即為「The number of matching array elements」。

Code – Bonus:



```
TITLE  hw4_bonus[hw4_bonus.asm]
INCLUDE Irvine32.inc

main EQU start@0

CountMatches PROTO,
    arr1:PTR SDWORD,
    arr2:PTR SDWORD,
    count:DWORD

.data
    array1 SDWORD 10,5,4,-6,2,11,12
    array2 SDWORD 10,5,3,1,4,2,-6

.code
CountMatches PROC,
    arr1: PTR SDWORD, arr2: PTR SDWORD, count: DWORD
    mov esi, arr1
    mov eax, 0
    mov ecx, count
L1:
    push ecx        ;nested loop
    mov ebx,[esi]
    mov ecx, count
    mov edi, arr2   ;point to the first element in arr2
L2:
    mov edx,[edi]
    cmp ebx,edx
    jne L3
    inc eax
L3:
    add edi, 4      ;point to the next element in arr2
    LOOP L2
    pop ecx

    add esi, 4      ;point to the next element in arr1
    LOOP L1
    ret
CountMatches ENDP

main PROC
    INVOKE CountMatches, ADDR array1, ADDR array2, LENGTHOF array1
main ENDP
END main
```

Result – Bonus:



```
TITLE  hw4_bonus[hw4_bonus.asm]
INCLUDE Irvine32.inc

main EQU start@0

CountMatches PROTO,
        arr1:PTR SDWORD,
        arr2:PTR SDWORD,
        count:DWORD

.data
        array1 SDWORD 10,5,4,-6,2,11,12
        array2 SDWORD 10,5,3,1,4,2,-6

.code
CountMatches PROC,
        arr1: PTR SDWORD, arr2: PTR SDWORD, count: DWORD
        mov esi, arr1
        mov eax, 0
        mov ecx, count
L1:
        push ecx                ;nested loop
        mov ebx,[esi]
        mov ecx, count
        mov edi, arr2   ;point to the first element in arr2
L2:
        mov edx,[edi]
        cmp ebx,edx
        jne L3
        inc eax
L3:
        add edi, 4              ;point to the next element in arr2
        LOOP L2
        pop ecx

        add esi, 4              ;point to the next element in arr1
        LOOP L1
        ret
CountMatches ENDP

main PROC
        INVOKE CountMatches, ADDR array1, ADDR array2, LENGTHOF array1
main ENDP
END main
```

```
EAX   00000005
EBX   0000000c
ECX   00000000
EDX   fffffffa
ESI   0040401c
EDI   00404038
EIP   0040105a
ESP   0019ff84
EBP   0019ff94
EFL   00000202
CS    0023
DS    002b
ES    002b
SS    002b
FS    0053
GS    002b

EA
```

Explanation – Bonus:

在 CountMatches 中，先把 eax 設為零以便於存放結果，並透過 stack 概念建立巢狀迴圈，此巢狀迴圈會將 arr1 和 arr2 所有元素倆倆比對，如果值相等，就會執行 inc eax 使 eax 加一，否則直接跳到 L3，CountMatches PROC 結束後，eax 中即為「The count of all the matching elements」。