Conditional Processing

Programming Assignment # 2

By

407261128

康智絜

X86 Assembly Language

Fall 2019

Date Submitted: November 02, 2019

**Test Plan :**

Analyze data stored in myID. The count of oddnumbers is stored inmyID\_oddCount. Sum up the evennumber in myID and store the value in myID\_evenSum using conditional jump. Multiply myID\_oddCount and myID\_evenSum and store in myID\_result.

myID = 407261128

myID\_oddCount = 26

myID\_evenSum = 5

myID\_result = 130

Output: 130

**Feedback: (required)**

1. 在if…else的地方想很久，最後是直接照著課本的格式打才成功

2. 常常打錯字 例如：jne打成jze 但是visual studio不會有特別的顏色提醒你打錯字，檢查很久才發現

3. 在判斷第i個字 -> if(i%2==0) 的部分找課本找很久，都沒有看到mod之類的用法，最後在網路上找到如果要mod一個數字，即mod x，只要將AND i, (x-1)就可以了，例如：i % 8 -> AND i, 7

4.除法的部分因為還沒有教到，但有找到在課本的288頁有介紹MUL，寫了很久一直是錯的，才發現

mov al, 5h

mov bl, 10h

mul bl ; al \* bl 的答案會存在 al 而不是bl

所以最後的答案應該把 al mov到 myID\_result裡面

5. WriteDec是印出無號數

WriteInt是印出有號數，所以數字前面會印出’+’號

**Appendix A: Test Log (required)**

Input Values : none

Date & Time: 2019/11/2 4:23pm

Actual Output : 130

Result : 130

**Appendix B: Source Code (required)**

INCLUDE Irvine32.inc

.data

myID BYTE 4,0,7,2,6,1,1,2,8

sizeID = ($-myID)

myID\_oddCount BYTE 0

myID\_evenSum BYTE 0

myID\_result BYTE 0

.code

main proc

mov ecx, sizeID

mov esi, 0

L1:

mov ebx, ecx ; ebx = 第幾個字

and ebx, 1 ; ebx = 第幾個字%2

cmp ebx, 0 ; if(ebx%2==0)

jne L2 ; if(ebx%2!=0) jump to L2

mov al, myID[esi] ; al = 第esi個字的數字 (ebx%2==0)

add myID\_evenSum, al ; myID\_evenCount += 第esi個字的數字 (ebx%2==0)

jmp L3

L2:

mov al, myID[esi]

add myID\_oddCount, al

jmp L3

L3:

add esi, TYPE BYTE ; esi += 1 BYTE

mov eax, esi

loop L1

mov al, myID\_oddCount

mov bl, myID\_evenSum

mul bl ; al = al \* bl

mov myID\_result, al ; myID\_result = al

call WriteDec ; print myID\_result

invoke ExitProcess,0

main endp

end main

