**#1**

INCLUDE Irvine32.inc

.data

bigEndian BYTE 12h,34h,56h,78h

littleEndian DWORD ?

.code

main PROC

mov esi ,offset bigEndian ;starting

mov edi, OFFSET bigEndian + SIZEOF bigEndian - TYPE bigEndian ;ending

mov ecx, LENGTHOF bigEndian ;counter=length

;display memory1

call dumpmem

mov eax,DWORD PTR bigEndian ;

mov littleEndian,eax

call writehex ;display bigEndian

call crlf

mov esi ,offset bigEndian

mov edi, OFFSET bigEndian + SIZEOF bigEndian - TYPE bigEndian

mov ecx, LENGTHOF bigEndian / 2 ;counter=4/2

L1: ;反轉陣列

push ecx

mov ecx, TYPE bigEndian

L2:

mov al,[esi]

mov ah,[edi]

mov [edi],al

mov [esi],ah

inc esi

inc edi

loop L2

pop ecx

SUB edi,2 \* TYPE bigEndian

loop L1

;display memory2

mov esi ,offset bigEndian ;starting

mov ecx, LENGTHOF bigEndian ;counter=length

call dumpMem

mov eax,DWORD PTR bigEndian

call writehex ;display littleEndian

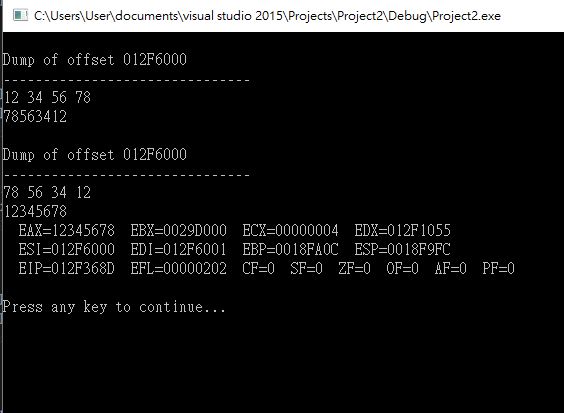
call DumpRegs

call WaitMsg

INVOKE ExitProcess ,0

main ENDP

END main



#2

INCLUDE Irvine32.inc

.data

array Byte 01h,02h,03h,04h

.code

main PROC

mov esi, offset array

mov ecx,LENGTHOF array

mov ebx,TYPE array

call DumpMem

mov ecx,LENGTHOF array/2

L1:

mov al,[esi]

mov ah,[esi+1]

mov [esi] ,ah

mov [esi+1], al

add esi,TYPE array

loop L1

mov esi,OFFSET array

mov ecx,LENGTHOF array

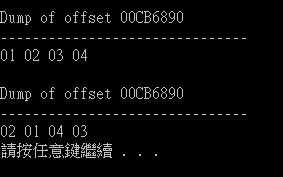
mov ebx,TYPE array

call DumpMem

INVOKE ExitProcess ,0

main ENDP

END main



**#3**

INCLUDE Irvine32.inc

.data

array dword 1,3,5,7

.code

main PROC

mov esi,offset array ;starting element

mov edi,offset array+sizeof array-type array ;endinb element

mov ecx,lengthof array-1 ;setting counter

mov edx,0 ;sum=0

L1: ;把差值用loop相加起來

mov eax,[esi]

mov ebx,[esi+type array]

sub ebx,eax

mov [esi],ebx

add edx,ebx

add esi ,type array

Loop L1

mov [edi],edx ;和給最後

;display

mov esi,OFFSET array

mov ecx,LENGTHOF array

mov ebx,TYPE array

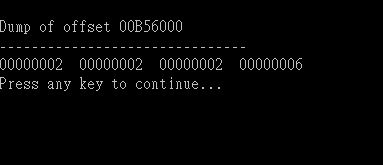
call DumpMem

call WaitMsg

exit

main ENDP

END main



**#5**

INCLUDE Irvine32.inc

.code

main PROC

mov ebx,1

mov eax,0

mov ecx,7

L1:

add eax,ebx

call DumpRegs ;display register

XCHG eax,ebx ; exchange eax and ebx

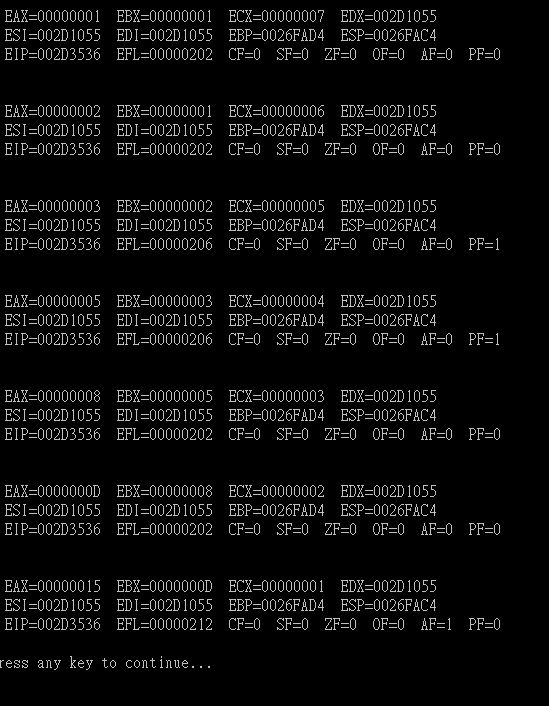
loop L1

call WaitMsg

exit

main ENDP

END main



#7

INCLUDE Irvine32.inc

.data

source BYTE "This is the source string",0

target BYTE SIZEOF source DUP('#')

.code

main PROC

mov esi,OFFSET source

mov edi,OFFSET target + LENGTHOF source -1

mov ecx,LENGTHOF source

call dumpmem

L1:

mov al,[esi]

mov [edi],al

inc esi

dec edi

loop L1

mov esi,OFFSET target+1

mov ebx,1

mov ecx,sizeOF target-1

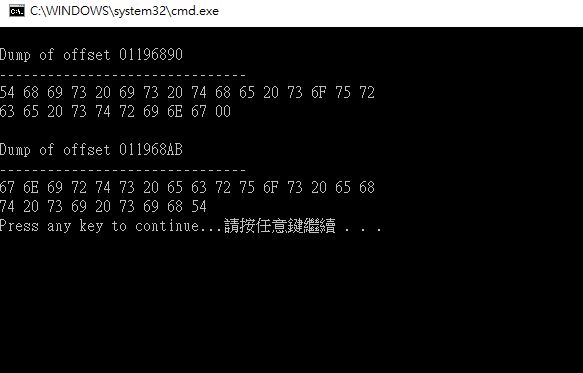
call dumpmem

call WaitMsg

exit

main ENDP

END main



#8

INCLUDE Irvine32.inc

.data

array Dword 1d,2d,3d,4d

.code

main PROC

mov esi,OFFSET array ;starting offset

mov ecx,LENGTHOF array ;number of unit

mov ebx,TYPE array ;double word format

call DumpMem

;display orignal order

mov esi,OFFSET array ;point to first element

mov edi,OFFSET array+sizeof array-type array ;point to last element

mov ecx,LENGTHOF array -1 ;setting couter

;setting

mov edx,[edi] ;last element give to edx

L1: ;每一個往後退

mov eax,[edi-type array]

mov [edi],eax

sub edi,type array

loop L1

mov [esi],edx ; edx give to first element

;display

mov esi,OFFSET array

mov ecx,LENGTHOF array

mov ebx,TYPE array

call DumpMem

call WaitMsg

exit

main ENDP

END main

