# Project 3 Report

Project 3 asked to reverse the array without copy the element to another one, also use 'SIZEOF', 'LENGTHOF', and 'TYPE' to calculate the array measurement in order to make flexible code. I use array's size, length and its element's size to calculate the rare pointer's position and assign the position to EDI. Also, I assign ESI to 0 in order to iterate the array from the front. In the loop, I exchange data with the one ESI pointed and EDI pointed, after that, I increase ESI to point its next element, and decrease EDI to point its previous element. After the loop, I use DumpMem to display the inversed array in memory.

Source Code:

TITLE Project 3 (p3.asm)

; Program Description: Display reverse array in memory

; Author: Chen Zhang

; Creation Date: 10/16/2019

; Revisions:

; Date: Modified by:

INCLUDE Irvine32.inc

.data

array DWORD 1,2,3,4,5,6,7,8,9,10,11

.code

main PROC

mov esi, 0

mov eax, SIZEOF array

mov ebx, TYPE array

mov edi, (SIZEOF array - TYPE array)

mov ecx, LENGTHOF array/2

call DumpRegs

;mov esi, OFFSET array

;mov ebx, 1

;mov ecx, SIZEOF array

;call DumpMem

L1:

mov eax, array[esi]

xchg eax, array[edi]

mov array[esi], eax

add esi, TYPE array

sub edi, TYPE array

loop L1

mov esi, OFFSET array

mov ebx, 1

mov ecx, SIZEOF array

call DumpMem

exit

main ENDP

END main

Output:

