Part 1 Reverse an array: <https://youtu.be/2wtjndyW8go>

Part 2 Reverse a string: <https://youtu.be/M7Fy9hCK9t0>

Part 3 Shift elements in array: <https://youtu.be/7CT2dOmEQlQ>

; Program Template (labw14p1.asm)

; Program Description: Reverse an array of integers

; Author: Timothy Bryant

; Creation Date: 4/24/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

dArray DWORD 1, 2, 3, 4, 5, 6, 7, 8

.code

main PROC

;write your code here

;get the addresses of the array

mov esi, OFFSET dArray

mov edi, OFFSET dArray

;add the array and the number of bytes used by array

add edi, SIZEOF dArray

;subtract the array by the size of each element the array

sub edi, TYPE dArray

;counter for loop

mov ecx, LENGTHOF dArray

;loop

restart:

;move the elements of the array in registers

mov eax, [esi]

mov ebx, [edi]

;swap the contents of the registers

xchg eax, ebx

;move the swapped elements back in the array

mov [esi], eax

mov [edi], ebx

;move on to the next element in the array

add esi, TYPE dArray

;move backwards in the array

sub edi, TYPE dArray

;decrement ecx-has length of array

dec ecx

loop restart

;get address for the array

mov esi, OFFSET dArray

;counter

mov ecx, LENGTHOF dArray

print:

;move element into eax

mov eax, [esi]

;display element

call WriteDec

;move onto the next element

add esi, TYPE dArray

loop print

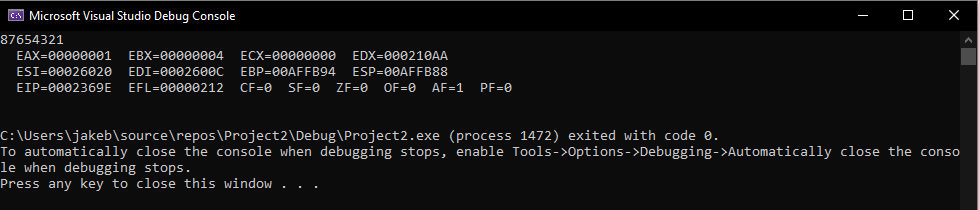
call DumpRegs

INVOKE ExitProcess, 0

main ENDP

; (insert additional procedures here)

END main



; Program Template (labw14p2.asm)

; Program Description: Reverse the order of string

; Author: Timothy Bryant

; Creation Date: 4/25/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

source BYTE "This is the source string", 0

target BYTE SIZEOF source DUP('#')

.code

main PROC

;write your code here

;move 0 into register to start at the begining of the string

mov esi, 0

;move the length of string minus 1 to start at end of the string

mov edi, SIZEOF source - TYPE source

;move size of string for loop in register

mov ecx, LENGTHOF source + 24

;loop

restart:

mov al, source[esi]

mov target[edi], al

;move on to the next element in the array

inc esi

dec edi

;decrement the size of string

dec ecx

loop restart

;display string in reverse order

mov edx, OFFSET target

call WriteString

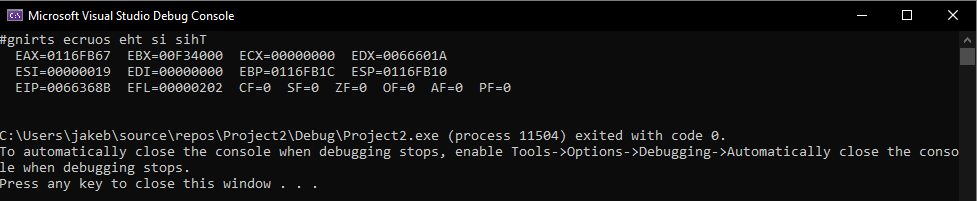
call DumpRegs

INVOKE ExitProcess, 0

main ENDP

; (insert additional procedures here)

END main



; Program Template (labw14p3.asm)

; Program Description: Shift elements in an Array

; Author: Timothy Bryant

; Creation Date: 4/27/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

dArray DWORD 10, 20, 30, 40

newArray DWORD LENGTHOF dArray DUP(?)

lastElement DWORD ?

.code

main PROC

;write your code here

;move the address of the first element

mov esi, OFFSET dArray

;move the address of the second element

mov edi, OFFSET newArray

add edi, TYPE newArray

;loop counter

mov ecx, LENGTHOF dArray - 1

restart:

;move the first element into eax

mov eax, [esi]

;move the first element into the second element in edi array

mov [edi], eax

;move onto the next elements

add esi, TYPE dArray

add edi, TYPE newArray

loop restart

;set last element in dArray to first in newArray

mov edi, OFFSET newArray

mov eax, [esi]

mov [edi], eax

;get address for first element

mov edi, OFFSET newArray

;set counter for print loop

mov ecx, LENGTHOF newArray

print:

;move the first element into eax

mov eax, [edi]

;display the element

call WriteDec

;move onto next element

add edi, TYPE newArray

loop print

call DumpRegs

INVOKE ExitProcess, 0

main ENDP

; (insert additional procedures here)

END main

