```
;-------------------------------------------------------------------------------
---
; Description: Demonstates the behaviors of stack, push and pop instructions
; Author name: Koichi Nakata
; Author email: kanakta595@insite.4cd.edu
; Last modified date: March 7, 2024
; Creation date: March 7, 2024
;-------------------------------------------------------------------------------
---


INCLUDE Irvine32.inc

.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode: dword

.data
aVal dword 121212h
bVal dword 343434h

.code
main PROC
        mov eax, aVal
        mov ebx, bVal
        call DumpRegs

        call exchangeEaxAndEbx
        call DumpRegs

        INVOKE ExitProcess, 0
main ENDP

;-------------------------------------------------------------------------------
---
; exchangeEaxAndEbx
; Exchanges values in EAX and EBX registers, only using push and pop instructions
; Receives: EAX = some 8-byte value
;                       EBX = some 8-byte value
; Returns: void
;-------------------------------------------------------------------------------
---
exchangeEaxAndEbx PROC
        push eax        ; stack = {eaxVal} -> top
        push ebx        ; stack = {eaxVal, ebxVal} -> top

        pop      eax    ; stack = {eaxVal}, ebxVal popped to eax
        pop  ebx        ; stack = {}, eaxVal popped to ebx
        ret                     ; Pops return address to eip
```

```
exchangeEaxAndEbx ENDP

END main
```

```
;-------------------------------------------------------------------------------
---
; Description: Manipulates return address using push and pop instructions
; Author name: Koichi Nakata
; Author email: kanakta595@insite.4cd.edu
; Last modified date: March 7, 2024
; Creation date: March 7, 2024
;-------------------------------------------------------------------------------
---

INCLUDE Irvine32.inc

.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode: dword

.data
aVal dword 121212h
bVal dword 343434h

.code
main PROC
        mov eax, aVal
        mov ebx, bVal

        call addNumAndLeap

        inc eax                         ; 1 byte - skipped
        inc eax                         ; 1 byte - skipped
        inc eax                         ; 1 byte - skipped
        inc eax                         ; 1 byte - skipped
```

```
        dec eax                         ; CPU returns here after addNumAndLeap
                                                ; EAX = 464645h instead of 464649h
                                                ; because inc instructons were
skipped

        INVOKE ExitProcess, 0
main ENDP

;-----------------------------------------------------------------------------
---
; addNumAndLeap
; Adds two numbers and returns to 4-byte higher address
; Receives: EAX = some 8-byte number
;                       EBX = some 8-byte number
; Returns: EAX = sum of the two numbers
;-----------------------------------------------------------------------------
---
addNumAndLeap PROC
        pop     esi             ; Save return address to esi
        add     esi, 4          ; Add 4 bytes to the return address
        push esi                ; Push back the address

        push ebx                ; Save the original value of ebx

        add  eax, ebx

        pop  ebx                ; Restore the original value of ebx
        ret
addNumAndLeap ENDP

END main
```