

HOMWORK 4 SOLUTIONS

QUESTIONS 4–10: ANALYSIS OF MEMORY ACCESSES

```

4.  main PROC          ; Display the last element of arr1
    mov esi, arr1      ESI = 1
    mov ecx, LENGTHOF arr1 ECX = 3
    call lastElement
    call WriteHex
    exit
main ENDP

lastElement PROC      1 + 3*4 - 4 = 9
    mov eax, DWORD PTR [esi + ecx*SIZEOF SDWORD - SIZEOF SDWORD]
    ret
lastElement ENDP

```

- If start = 00401000h, what value is passed in ESI to the lastElement procedure? *(00000001h)*
- Consider the memory operand for the mov instruction in the lastElement procedure. If start = 00401000h, what memory address does it access? *00000009h*
- The following image shows the 48 bytes of the memory, beginning at 00401000h. start = 00401000h, what four bytes comprise the DWORD value that is copied into EAX? Circle them. If the memory address that is accessed is not shown, write "Not shown."

Memory 1									
Address: 0x00401000									
0x00401000	00	00	00	01	00	00	00	02
0x00401008	00	00	00	03	00	00	00	ddY
0x00401010	cc	bb	aa	cc	dd	ee	ff	00	i»=iYiy.
0x00401018	00	00	00	00	00	00	00	00
0x00401020	01	30	31	32	33	34	35	36	.0123456
0x00401028	37	38	39	41	42	43	44	45	789ABCDE

Not shown

- For every program, the expected output is 00000003—the last value in the arr1 array.. What will the *actual* output be? Or will the program crash due to an invalid memory access?

Crash

```

5.  main PROC          ; Display the last element of arr1
    mov esi, OFFSET arr1    ESI = 00401003h
    mov ecx, LENGTHOF arr1  ECX = 3
    call lastElement
    call WriteHex
    exit
main ENDP

lastElement PROC
    401003h + 3*4 = 40100Fh
    mov eax, DWORD PTR [esi + ecx*SIZEOF SDWORD]
    ret
lastElement ENDP

```

- If start = 00401000h, what value is passed in ESI to the lastElement procedure? *00401003h*
- Consider the memory operand for the mov instruction in the lastElement procedure. If start = 00401000h, what memory address does it access? *0040100Fh*
- The following image shows the 48 bytes of the memory, beginning at 00401000h. start = 00401000h, what four bytes comprise the DWORD value that is copied into EAX? Circle them. If the memory address that is accessed is not shown, write "Not shown."

Memory 1									
Address:		0x00401000							
0x00401000	00	00	00	01	00	00	00	02
0x00401008	00	00	00	03	00	00	00	<u>dd</u>Ÿ
0x00401010	<u>cc</u>	<u>bb</u>	<u>aa</u>	cc	dd	ee	ff	00	ix=İŸiy.
0x00401018	00	00	00	00	00	00	00	00
0x00401020	01	30	31	32	33	34	35	36	.0123456
0x00401028	37	38	39	41	42	43	44	45	789ABCDE

- For every program, the expected output is 00000003—the last value in the arr1 array.. What will the *actual* output be? Or will the program crash due to an invalid memory access?

AABBCCDD

```

7.  main PROC          ; Display the last element of arr1
    mov esi, OFFSET arr1
    mov ecx, LENGTHOF arr1
    call lastElement
    call WriteHex
    exit
main ENDP

```

```

lastElement PROC
    mov eax, DWORD PTR [esi + ecx - SIZEOF SDWORD]
    ret
lastElement ENDP

```

- a. If start = 00401000h, what value is passed in ESI to the lastElement procedure? 00401003h
- b. Consider the memory operand for the mov instruction in the lastElement procedure. If start = 00401000h, what memory address does it access? 00401002h
- c. The following image shows the 48 bytes of the memory, beginning at 00401000h. start = 00401000h, what four bytes comprise the DWORD value that is copied into EAX? Circle them. If the memory address that is accessed is not shown, write "Not shown."

Memory 1									
Address:		0x00401000							
0x00401000	00	00	00	01	00	00	00	02
0x00401008	00	00	00	03	00	00	00	ddÝ
0x00401010	cc	bb	aa	cc	dd	ee	ff	00	İ»=İÝiy.
0x00401018	00	00	00	00	00	00	00	00
0x00401020	01	30	31	32	33	34	35	36	.0123456
0x00401028	37	38	39	41	42	43	44	45	789ABCDE

- d. For every program, the expected output is 00000003—the last value in the arr1 array.. What will the *actual* output be? Or will the program crash due to an invalid memory access?

00000100


```

9.  main PROC          ; Display the last element of arr1
    mov esi, OFFSET arr1
    mov ecx, LENGTHOF arr1
    call lastElement
    call WriteHex
    exit
main ENDP

```

```

lastElement PROC
    mov eax, DWORD PTR [esi + ecx*SIZEOF SDWORD - 2*SIZEOF SDWORD]
    ret
lastElement ENDP

```

$00401003h$
 3
 $401003h + 3 * 4 - 8 = 401007h$

- If start = 00401000h, what value is passed in ESI to the lastElement procedure?
- Consider the memory operand for the mov instruction in the lastElement procedure. If start = 00401000h, what memory address does it access?
- The following image shows the 48 bytes of the memory, beginning at 00401000h. start = 00401000h, what four bytes comprise the DWORD value that is copied into EAX? Circle them. If the memory address that is accessed is not shown, write "Not shown."

Memory 1	
Address:	0x00401000
0x00401000	00 00 00 01 00 00 00 02
0x00401008	00 00 00 03 00 00 00 dd
0x00401010	cc bb aa cc dd ee ff 00
0x00401018	00 00 00 00 00 00 00 00
0x00401020	01 30 31 32 33 34 35 36
0x00401028	37 38 39 41 42 43 44 45

- For every program, the expected output is 00000003—the last value in the arr1 array.. What will the *actual* output be? Or will the program crash due to an invalid memory access?

00000002

10. main PROC ; Display the last element of arr1

mov esi, OFFSET arr1 *00401003h*

mov ecx, LENGTHOF arr1 *3*

call lastElement

call WriteHex

exit

main ENDP

lastElement PROC

mov eax, DWORD PTR [ecx + esi*SIZEOF SDWORD - SIZEOF SDWORD]

ret

lastElement ENDP

$$3 + 401003h * 4 - 4 = 0100400B$$

a. If start = 00401000h, what value is passed in ESI to the lastElement procedure? *00401003h*

b. Consider the memory operand for the mov instruction in the lastElement procedure. If start = 00401000h, what memory address does it access?

0100400Bh

c. The following image shows the 48 bytes of the memory, beginning at 00401000h. start = 00401000h, what four bytes comprise the DWORD value that is copied into EAX? Circle them. If the memory address that is accessed is not shown, write "Not shown."

Memory 1									
Address:		0x00401000							
0x00401000	00	00	00	01	00	00	00	02
0x00401008	00	00	00	03	00	00	00	ddÝ
0x00401010	cc	bb	aa	cc	dd	ee	ff	00	İ»=İÝiy.
0x00401018	00	00	00	00	00	00	00	00
0x00401020	01	30	31	32	33	34	35	36	.0123456
0x00401028	37	38	39	41	42	43	44	45	789ABCDE

Not shown

d. For every program, the expected output is 00000003—the last value in the arr1 array.. What will the *actual* output be? Or will the program crash due to an invalid memory access?

Crash