





|      |        |
|------|--------|
| var1 | 65 'A' |
| var2 | 66 'B' |
| var3 | 67 'C' |
| var4 | 68 'D' |



Memory 1 Registers Watch 1

peter\_ruszel\_260\_assign5.asm

```
1  .386
2  .model flat,stdcall
3  .stack 4096
4  ExitProcess proto,dwExitCode:dword
5
6  .data
7      var1 BYTE 'A'
8      var2 BYTE 'B'
9      var3 BYTE 'C'
10     var4 BYTE 'D'
11
12  .code
13  main proc
14      mov AH, var1
15      mov AL, var2
16      mov var2, AH
17      mov AH, var3
18      mov var3, AL
19      mov AL, var4
20      mov var4, AH
21      mov var1, AL
22      invoke ExitProcess,0
23  main endp
24  end main
```

|  |        |
|--|--------|
|  var1   | 68 'D' |
|  var2   | 65 'A' |
|  var3  | 66 'B' |
|  var4 | 67 'C' |

Memory 1   Registers   **Watch 1**

peter\_ruszel\_260\_assign5.asm    

```

1  .386
2  .model flat,stdcall
3  .stack 4096
4  ExitProcess proto,dwExitCode:dword
5
6  .data
7      var1 BYTE 'A'
8      var2 BYTE 'B'
9      var3 BYTE 'C'
10     var4 BYTE 'D'
11
12  .code
13  main proc
14      mov AH, var1
15      mov AL, var2
16      mov var2, AH
17      mov AH, var3
18      mov var3, AL
19      mov AL, var4
20      mov var4, AH
21      mov var1, AL
22      invoke ExitProcess,0
23  main endp
24  end main

```

≤ 1ms elapsed

Registers  
EAX = 44414243 EBX = 002D3000 ECX = 00401005 EDX = 00401005 ESI = 00401005 EDI = 0

100 %  
Memory 1 Registers Watch 1

peter\_ruszel\_260\_assign5.asm

```
26 mov ecx, 0
27
28
29
30 ; Phase 2
31 ; move first two bytes into lower 16-bit registers of EAX
32 mov AH, var1
33 mov AL, var2
34
35 ; shift first two bytes into upper 16-bits of EAX
36 add eax, eax
37 add eax, eax
38 add eax, eax
39 add eax, eax
40 add eax, eax
41 add eax, eax
42 add eax, eax
43 add eax, eax
44 add eax, eax
45 add eax, eax
46 add eax, eax
47 add eax, eax
48 add eax, eax
49 add eax, eax
50 add eax, eax
51 add eax, eax
52
53 ; move last two bytes into lower 16-bit registers of EAX
54 mov AH, var3
55 mov AL, var4
56
57
58
59 ; Phase 3
60
```

100 % No issues found

## Registers

EAX = 44414243 EBX = BCBFBEBD ECX = 00401005 EDX = 00401005 ESI = 0

100 %

Memory 1 Registers Watch 1

peter\_ruszel\_260\_assign5.asm

```
61      ; Phase 3
62      ;-----
63      ; reset EBX
64      mov ebx, 0
65
66      ; get two's complement of first two bytes
67      sub bh, var1
68      sub bl, var2
69
70      ; shift first two bytes into upper 16-bits of EBX
71      add ebx, ebx
72      add ebx, ebx
73      add ebx, ebx
74      add ebx, ebx
75      add ebx, ebx
76      add ebx, ebx
77      add ebx, ebx
78      add ebx, ebx
79      add ebx, ebx
80      add ebx, ebx
81      add ebx, ebx
82      add ebx, ebx
83      add ebx, ebx
84      add ebx, ebx
85      add ebx, ebx
86      add ebx, ebx
87
88      ; get two's complement of last two bytes
89      ; and move into lower 16-bit registers of EBX
90      sub bh, var3
91      sub bl, var4
92
93      invoke ExitProcess,0 ≤ 1ms elapsed
94  main endp
```

100 %

✓ No issues found