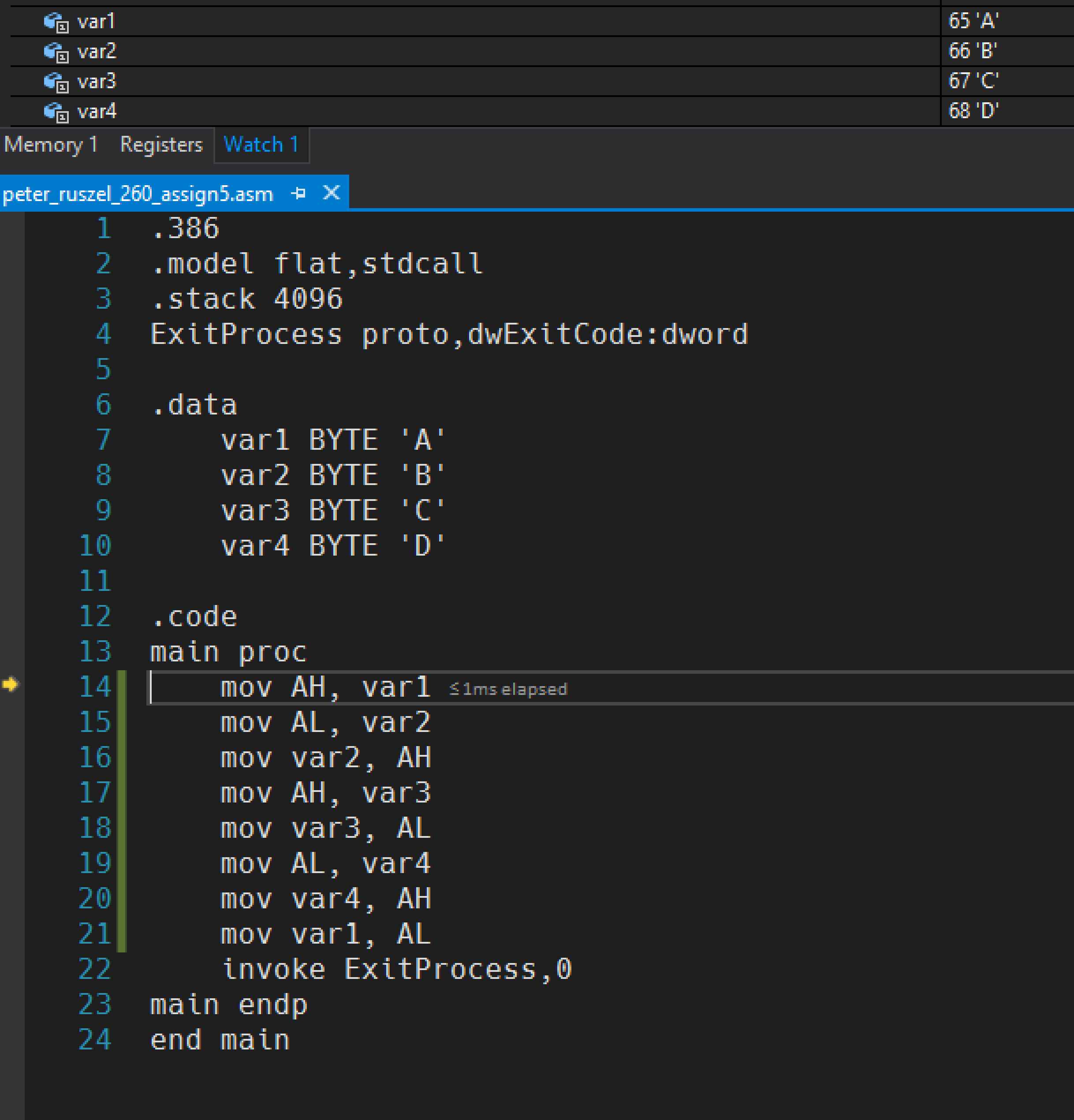
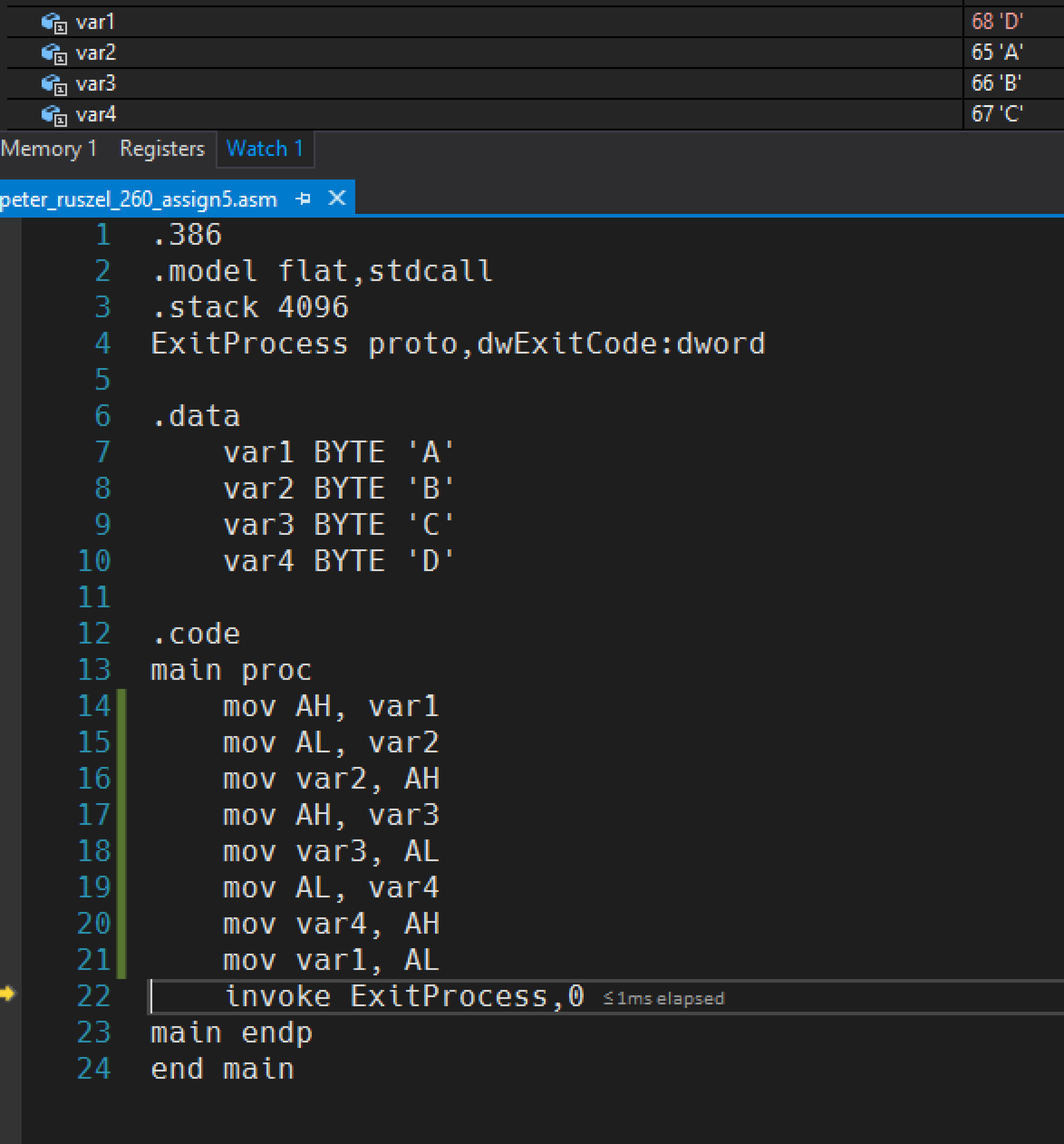


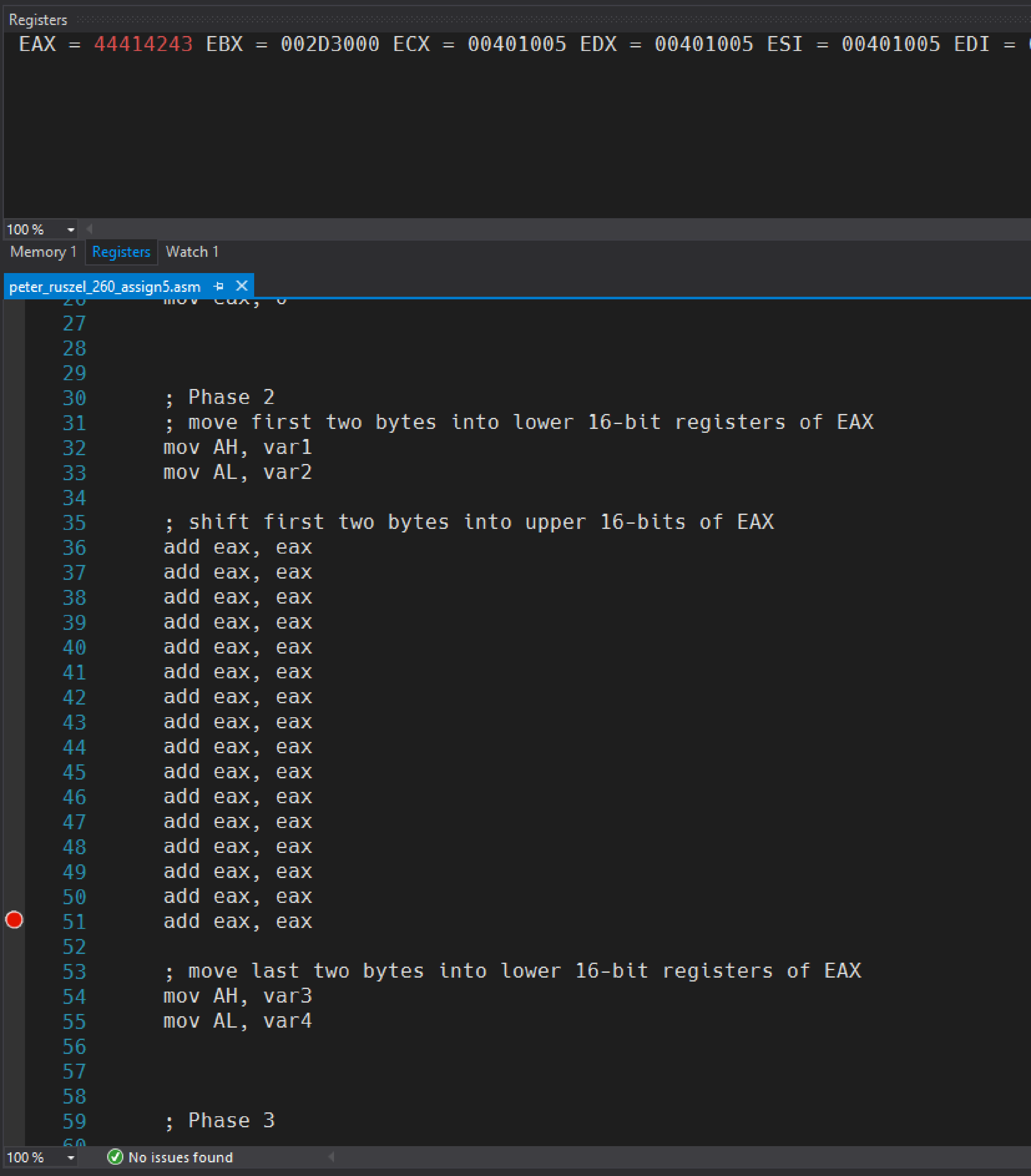
```
var1    65 'A'
var2    66 'B'
var3    67 'C'
var4    68 'D'
```

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

.data
    var1 BYTE 'A'
    var2 BYTE 'B'
    var3 BYTE 'C'
    var4 BYTE 'D'

.code
main proc
    mov AH, var1
    mov AL, var2
    mov var2, AH
    mov AH, var3
    mov var3, AL
    mov AL, var4
    mov var4, AH
    mov var1, AL
    invoke ExitProcess,0
main endp
end main
```



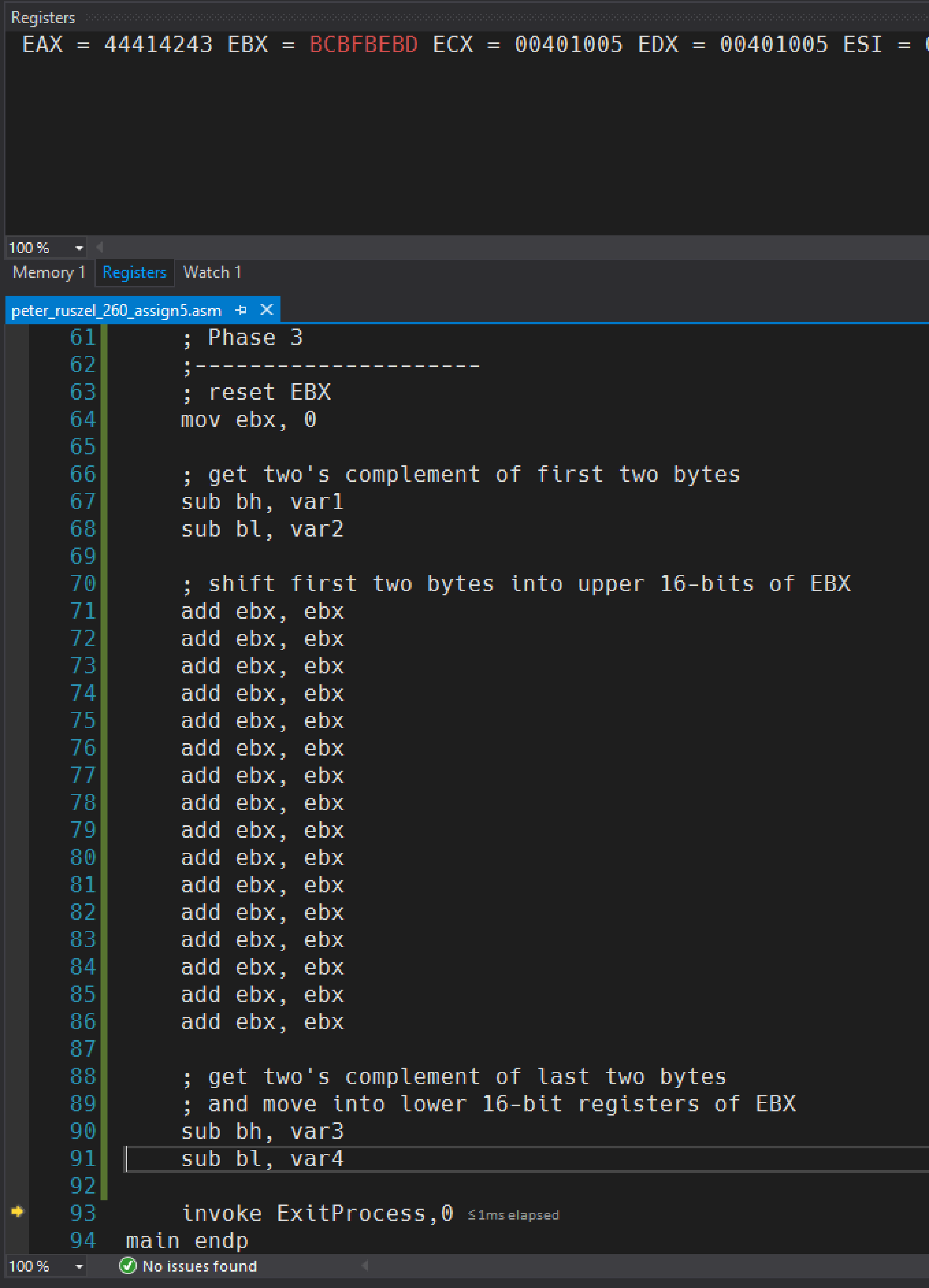
var1	68 'D'
var2	65 'A'
var3	66 'B'
var4	67 'C'

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

.data
    var1 BYTE 'A'
    var2 BYTE 'B'
    var3 BYTE 'C'
    var4 BYTE 'D'

.code
main proc
    mov AH, var1
    mov AL, var2
    mov var2, AH
    mov AH, var3
    mov var3, AL
    mov AL, var4
    mov var4, AH
    mov var1, AL
    invoke ExitProcess,0
main endp
end main
```



```
EAX = 44414243 EBX = 002D3000 ECX = 00401005 EDX = 00401005 ESI = 00401005 EDI =
```

```
        mov eax, 0



        ; Phase 2
        ; move first two bytes into lower 16-bit registers of EAX
        mov AH, var1
        mov AL, var2

        ; shift first two bytes into upper 16-bits of EAX
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax
        add eax, eax

        ; move last two bytes into lower 16-bit registers of EAX
        mov AH, var3
        mov AL, var4



        ; Phase 3
```



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

```
    ; Phase 3
    ;---------------------
    ; reset EBX
    mov ebx, 0

    ; get two's complement of first two bytes
    sub bh, var1
    sub bl, var2

    ; shift first two bytes into upper 16-bits of EBX
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx
    add ebx, ebx

    ; get two's complement of last two bytes
    ; and move into lower 16-bit registers of EBX
    sub bh, var3
    sub bl, var4

    invoke ExitProcess,0
main endp
```