

- from memory to register, remember that it is stored in memory in reverse byte order
- to reference memory, use brackets
 - mov ebx, offset wT ; address of wT
 - mov eax, [ebx] ; load address of ebx into eax
- | | | |
|----|-------|----|
| 24 | 37/AE | 57 |
|----|-------|----|

 - mov ax, wT
 - mov ax, wT + 1 AE37
 - mov ax, wT[1] - 1 reference byte, not position (offset)
- wT word 5, 4, 3, 2, 1, 0
 short wT = {5, 4, 3, 2, 1, 0}
 - mov ax, wT ax: 0005
 - mov ax, wT[1] is the same as mov ax, wT + 1 ax: 0000
 - mov ax, wT[4] ax: 0003
- mov eax, [ebx + esi] - add two registers to determine address
 must use ebx & esi - class standard
- ml /c /coff /Zi /Fl test.asm
 macro.bat
 " " %1.asm
- link /debug /subsystem:console /entry:start /out:%1.exe %1.obj
 convut.1201109.obj kernel32.lib utility201109.obj