Identify the problems in the following instructions and correct them by replacing them with one or two instruction having the same effect.

**a. mov bx, al**

The problem here is a **size mismatch**: you are trying to move an 8-bit value into a 16-bit register, which is not allowed.

Solution:

xor bh, bh ; Clear BH (upper byte of BX)

mov bl, al ; Move AL to BL (lower byte of BX)

**OR**

movzx bx, al ; Zero-extend AL into BX (this is valid in 32-bit or 64-bit architectures)

**b. mov ax, [si+di+100]**

The problem here is that in 16-bit real-mode addressing (like in x86), not all combinations of registers are allowed for effective addressing.

In **16-bit addressing modes**, you can use either:

* **Base Register** (BX or BP) with an index register (SI or DI), plus a displacement, like:

mov ax, [bx+si+100] ; Valid

mov ax, [bp+di+100] ; Valid