all it is -- data. In the second example, the C0 is a long branch instruction requiring 2 bytes of data. This data, which immediately follows the C0 instruction is an address - 0406 - and the program would jump to this address. If DISASSEMBLER-7 did not recognize this fact, it could possibly view data as instructions causing a listing which would be hopelessly confusing to read.

Therefore, DISASSEMBLER-7 contains a feature which makes it impossible to view from the middle of data. To illustrate what I mean, read in the first three pages of DISASSEMBLER-7 for disassembly, and follow along in the typed listing.

Notice that at ML 0009, we have the instruction F8FF which, as previously explained, means load the D register with the byte FF. Now press Key D to select SHOW FROM. Enter the address 000A and you will see that the program recognizes the byte at 000A to be data -- it is not an instruction -- so it displays from 0009 even though we requested 000A. In the event that two such instructions occur back to back, the listing will <u>always</u> begin with the first instruction it can be sure is not a piece of data to another instruction. (These data bytes, by the way, are commonly referred to as arguments.) Try -- but not just now -- to load a page with C0 C0 C0, long branches to the address C0 C0, then attempt to view from the middle