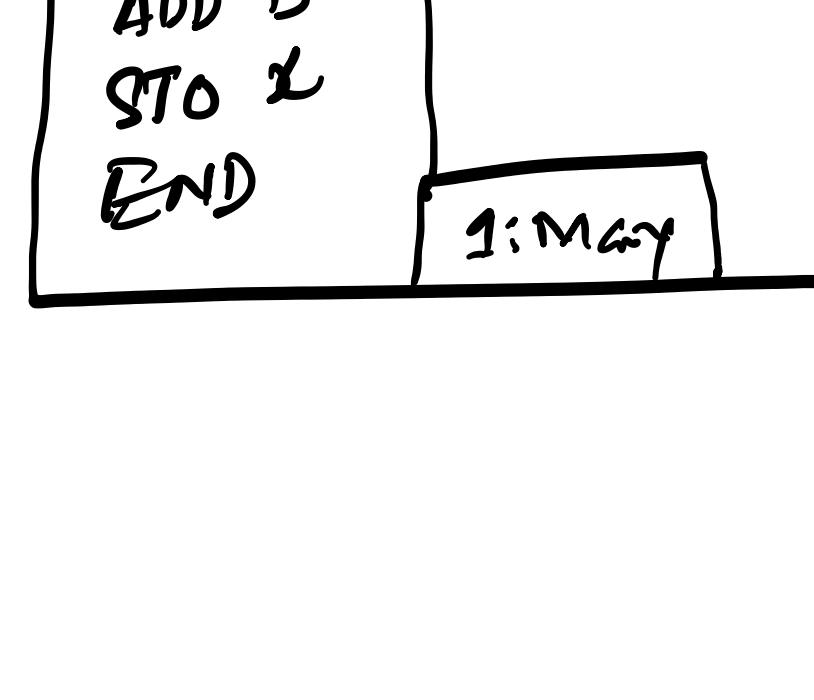
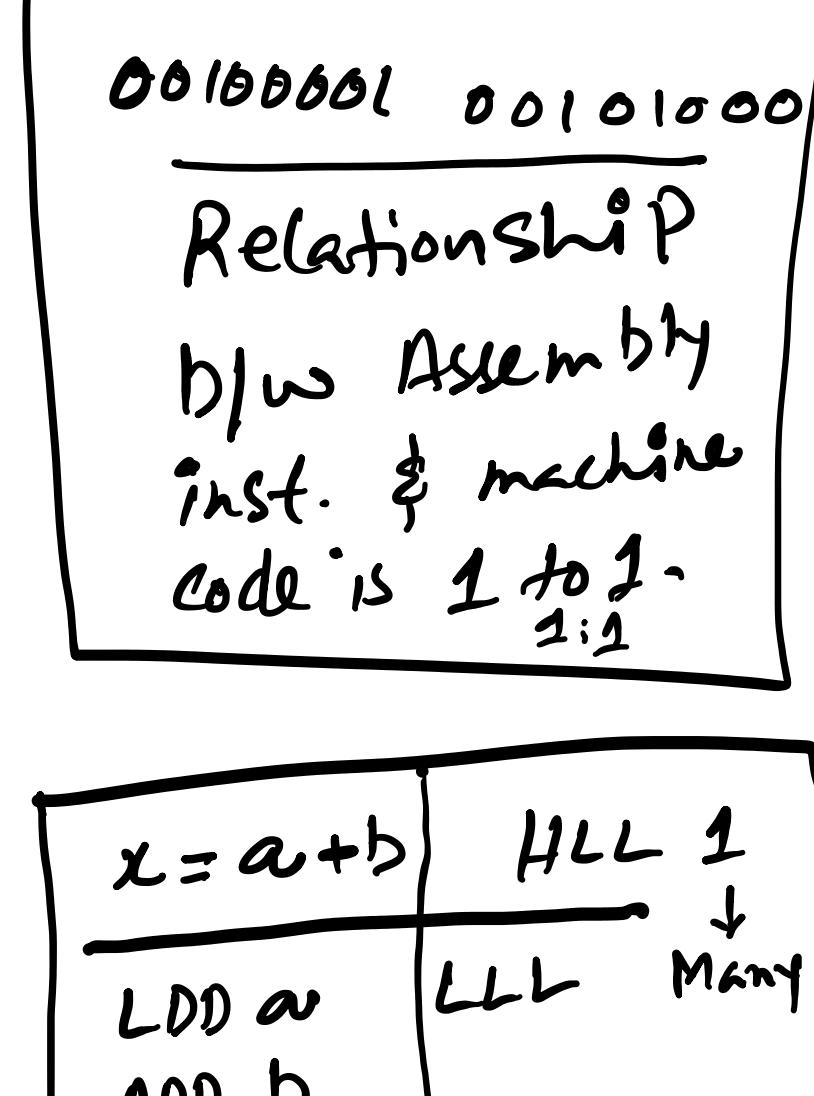


### Low level languages (LLL):

- They are called Assembly language.
- Their translators are 'Assemblers'.
- LLL are machine dependent.
- Every processor (family) has its own assembly language.
- Assembly language commands/instructions are simple and can't be broken further.
- Assembly instructions are called Mnemonics, OP-codes or Pseudo-commands.
- There is 1:1 relationship b/w assembly inst. & machine code.
- These are used to develop hardware related programs like Device drivers, operating system modules and gaming cores.
- Commands are hard to remember and are non-English like.



### High level Languages (HLL):

- These languages have English like syntax.
- These are easy to remember and read.
- These are many; like JAVA, PYTHON, C++ etc.
- Commands are complex and broken down into number of shorter instructions in machine language.
- Relationship b/w HLL and machine code are 1 to many.
- HLL are used to develop application programs; like accounting apps, web based apps and word processing apps etc.
- HLL are machine independent.

Why Compiler is memory hungry?  
Why compiler takes a lot of memory?

Because:

- Compiler loads itself.
- Compiler loads whole source code.
- Compiler keeps working space in memory.
- Machine language/code is also kept in memory after being developed/produced.

Memory (RAM) in early computers during 70's and later was expensive. Since, compilers were taking a lot of memory, they were not quite popular with programmers. To make programming popular among computer programmers, interpreters were made, which take less memory when they are converting source code to machine code.