 Provide examples of three different instruction mnemonics.

Step 1:

A binary machine instruction operation code is represented by a word or acronym in assembly language. Because different processors have different instruction sets, different mnemonics are used to represent them. Mnemonic instruction is a set of tactics that students can use to help them remember new information. Through the use of visual and/or aural clues, mnemonics education connects new information to existing knowledge.

Step 2:

Mnemonics that correlate to machine language instructions can be used to specify programme instructions in the Assembly language.

A mnemonic is an acronym for an operation in the computer assembler (or assembly) language. Each assembler programme instruction's operation code field contains it. AND AC, 37, for example, indicates AND the AC register with 37. As a result, so AND , **SUB and MUL** are mnemonic.

What is a calling convention, and how is it used in assembly language declarations?

Step 1:

A calling convention is a technique for how subroutines take parameters from their caller and provide a result at the implementation level (low level).

The way functions on a specific architecture and operating system interact is governed by a calling convention. This contains rules for where function arguments go, where return values go, what registers functions can utilise, how local variables are allocated, and so on.

Step 2:

The way parameters are supplied to subroutines and how the stack is restored after the subroutine call are determined by the calling convention.

The way arguments are supplied to a function, how return values are passed back out of a function, how the function is called, and how the function handles the stack and its stack frame are all defined by calling conventions. In a nutshell, the calling convention describes how a C or C++ function call is translated into assembly language.