Name: **Sahil Dattatray Mohite**

Roll No: **30** Batch: **B2**

PRN: 12010501

Lab Assignment 2

**Write an ALP program to accept 2 single digit number by using 01h INT21h function and display its addition using 02h INT21h function**

**AH = 01h**

* READ CHARACTER FROM STANDARD INPUT
* Return: AL = character read

**AH = 02h**

* WRITE CHARACTER TO STANDARD OUTPUT
* Entry: DL = character to write
* Return: AL = last character output

**add** — Integer addition

* It is arithmetic instruction.
* The add instruction adds together its two operands, storing the result in its second operand. Note, whereas both operands may be registers, at most one operand may be a memory location.
* *Syntax*  
  add <reg>, <reg>  
  add <mem>, <reg>  
  add <reg>, <mem>  
  add <con>, <reg>  
  add <con>, <mem>

**lea** — Load effective address

The lea instruction places the address specified by its first operand into the register specified by its second operand. Note, the contents of the memory location are not loaded, only the effective address is computed and placed into the register. This is useful for obtaining a pointer into a memory region or to perform simple arithmetic operations.

**cmp** — Compare

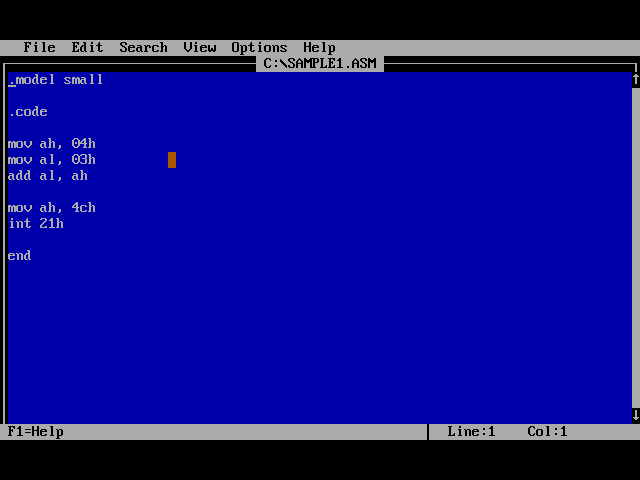
Compare the values of the two specified operands, setting the condition codes in the machine status word appropriately. This instruction is equivalent to the sub instruction, except the result of the subtraction is discarded instead of replacing the first operand.

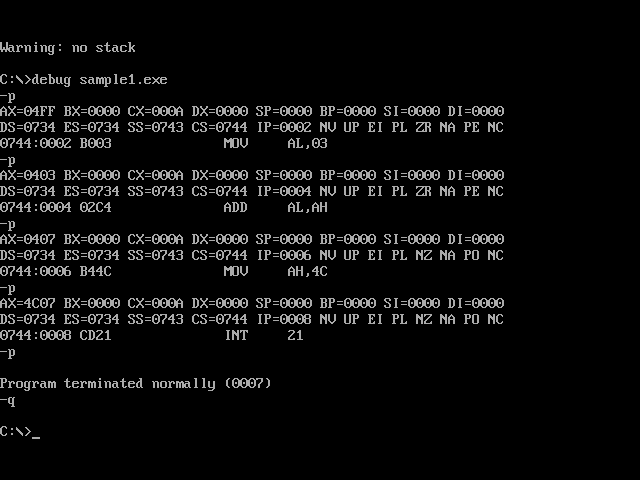
*Syntax*  
cmp <reg>, <reg>  
cmp <mem>, <reg>  
cmp <reg>, <mem>  
cmp <con>, <reg>

**j*condition*** — Conditional jump

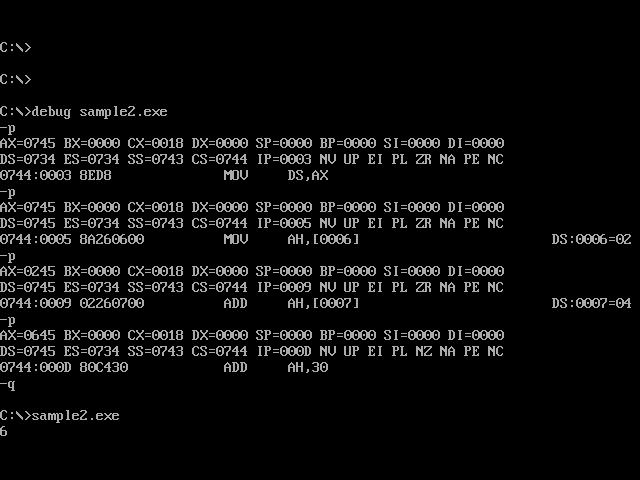
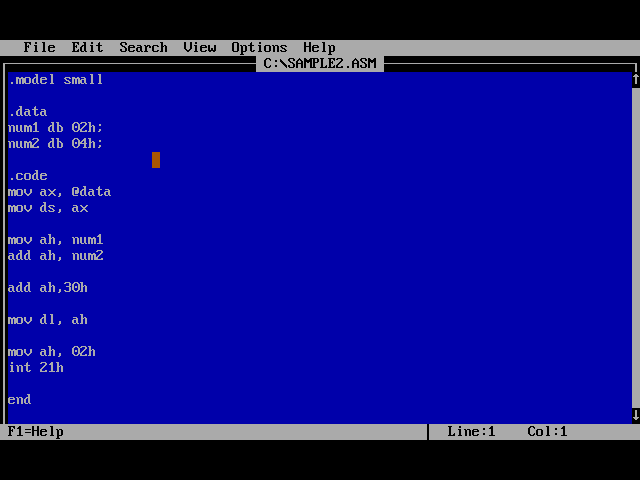
* These instructions are conditional jumps that are based on the status of a set of condition codes that are stored in a special register called the *machine status word*. The contents of the machine status word include information about the last arithmetic operation performed.
* A number of the conditional branches are given names that are intuitively based on the last operation performed being a special compare instruction, cmp (see below). For example, conditional branches such as jle and jne are based on first performing a cmp operation on the desired operands.
* *Syntax*  
  je <label> (jump when equal)  
  jne <label> (jump when not equal)  
  jz <label> (jump when last result was zero)  
  jg <label> (jump when greater than)  
  jge <label> (jump when greater than or equal to)  
  jl <label> (jump when less than)  
  jle <label> (jump when less than or equal to)

Sample 1





Sample 2



Sample 3

