COA – Lab Assignment 2

Name: Bhavin Patil

Roll No- 78

GR No : 12120056

Div : CS- D

Batch : B3

Problem Statement : to understand 01h and 02h functions of INT 21h

Instructions –

- MOV: This instruction is used to move data from one location to another.
  Syntax – mov destination, source

- LEA (Load Effective Address): It loads the specified register with the offset of a memory location.

- ADD: it performs an addition on both the first source register's contents and the second source. register's contents, and stores the result in the destination register

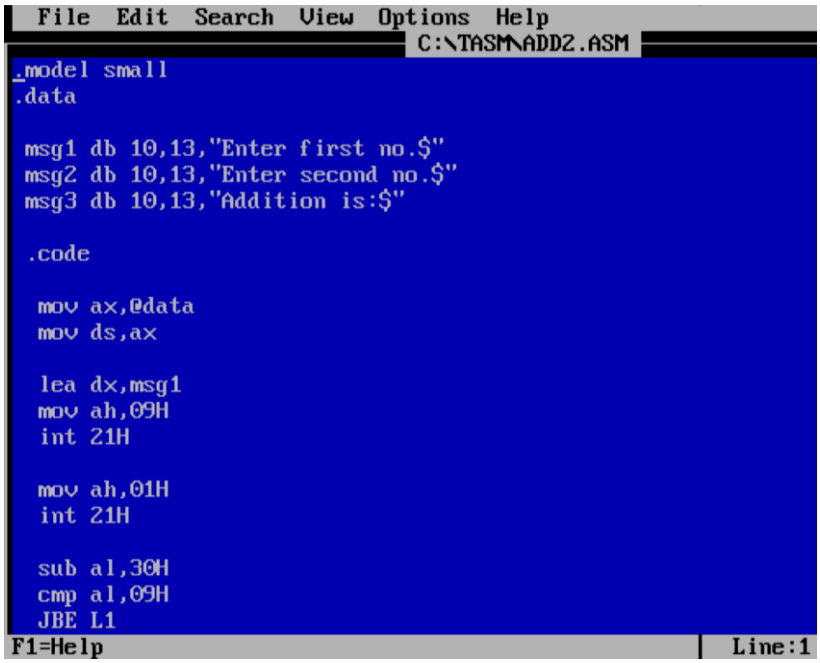
Commands –

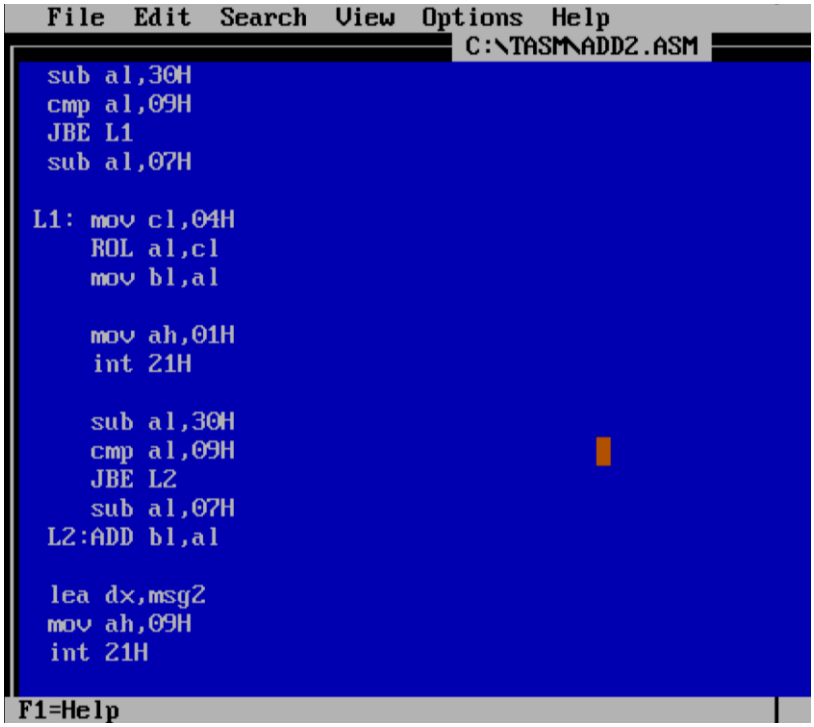
1. **01h :** It is used to read character from standard input, with echo, result is stored in AL.
2. **02h :** It is used to display single character
3. **09h**: Displays the string until "$" is reached.
4. **Int 21h**: Interrupt used to exit the program.
5. **.data**: This Command is used only when we want to store in Data Segment, basically, it is the memory access of the Data Segment. Whatever we want to print must be written here. Also, the variables are declared here.
6. **10, 13**: They work as Escape Sequence Character
7. **$**: It states the end of a Statement
8. **Db (Define Byte)**: It acts as an Assembler Directive
9. **. code**: Full Logical Program is written here

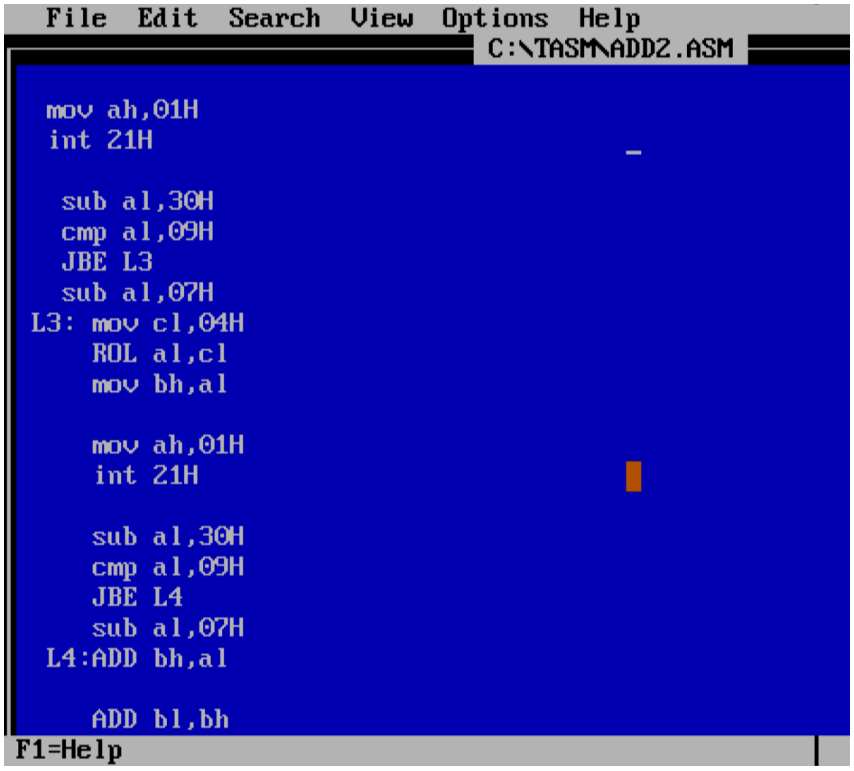
10. **Tasm –** Used for Compilation
11. **tlink**– Perform linking operation

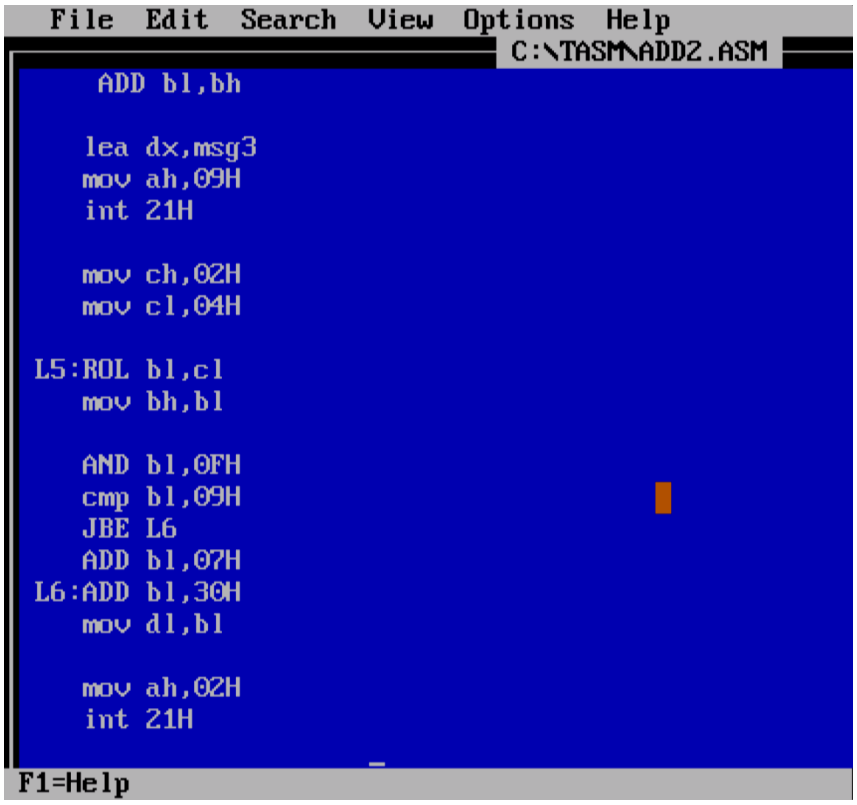
Screenshots of Source Code and Output:

Source Code -






File Edit Search View Options Help
C:\TASM\ADD2.ASM
mov ah,01H
int 21H
sub al,30H
cmp al,09H
JBE L3
sub al,07H
L3: mov cl,04H
ROL al,cl
mov bh,al
mov ah,01H
int 21H
sub al,30H
cmp al,09H
JBE L4
sub al,07H
L4:ADD bh,al
ADD bl,bh
F1=Help


File Edit Search View Options Help
C:\TASM\ADD2.ASM
ADD bl,bh
lea dx,msg3
mov ah,09H
int 21H
mov ch,02H
mov cl,04H
L5:ROL bl,cl
mov bh,bl
AND bl,0FH
cmp bl,09H
JBE L6
ADD bl,07H
L6:ADD bl,30H
mov dl,bl
mov ah,02H
int 21H
F1=Help

```
File  Edit  Search  View  Options  Help
                                C:\TASM\ADD2.ASM

L5:ROL bl,cl
   mov bh,bl

   AND bl,0FH
   cmp bl,09H
   JBE L6
   ADD bl,07H
L6:ADD bl,30H
   mov dl,bl

   mov ah,02H
   int 21H

   mov bl,bh
   dec ch
   JNZ L5

   mov ah,4ch
   int 21H
   end
```

Output:

```
C:\TASM>TASM ADD2.asm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   ADD2.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  475k


C:\TASM>tlink ADD2
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>add2.exe

Enter first no.12
Enter second no.13
Addition is:25
C:\TASM>

C:\TASM>
```