Eg :

Program for addition

1. Open notepad
2. .model small

.code

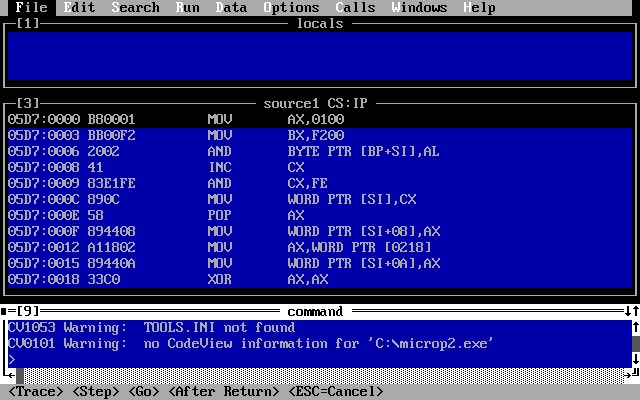
1. MOV AL, 10H

MOV BL, 10H

ADD AL, BL

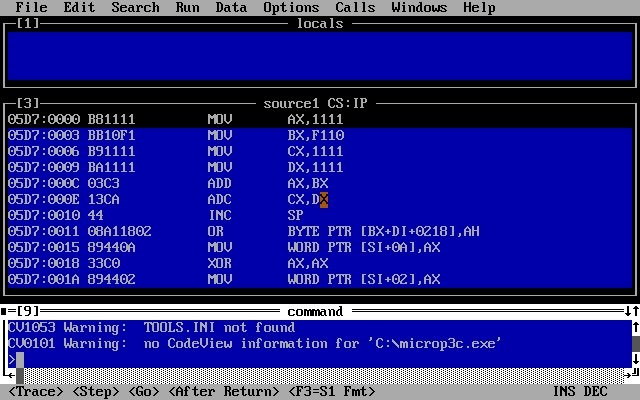
1. end
2. save file in bin folder of masm with the extension .asm
3. close the file
4. open command prompt
5. cd \
6. cd masm\BIN
7. ml filename.asm
8. open dos box
9. alt + enter – bigger size of the screen
10. mount C c:\masm\BIN
11. C:
12. cv filename.exe

WAP to change content of AX register to 0100H and BX register to F200H

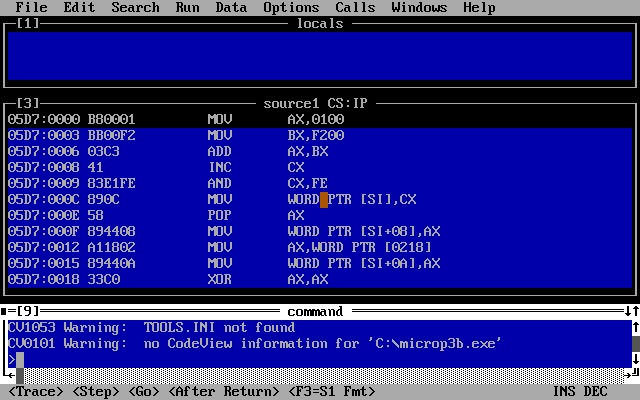


# WAP to perform 8 bit, 16 bit and 32 bit addition using 16 bit instruction mode.

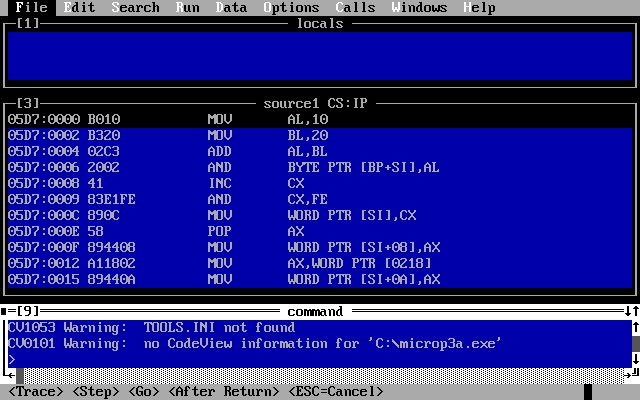
32 bit



16 bit



8 bit



# WAP to perform 8 bit,16 bit, 32 bit subtraction using 16 bit instruction mode.

# 32 bit

# 

# 16 bit

# 

# 8 bit

# 

# WAP to swap the 1st and 4th element of an array containing 5 word sized elements

(in folder - swap.asm)

# Write a program to perform 32 bit BCD addition and subtraction.

32 bit addition



32 bit subtraction



# Write a program to perform 8 and 16 bit multiplication and division.

8 bits division

Graphical user interface

Description automatically generated

16 bits division

Graphical user interface

Description automatically generated

8 bits multiplication

Graphical user interface

Description automatically generated

16 bits multiplication

Graphical user interface

Description automatically generated

# Wap to convert 16 bit binary content of AX into 4 digit ASCII character string

Graphical user interface

Description automatically generated

# Wap to perform Binary to ASCII conversion.

Graphical user interface

Description automatically generated

# Wap to perform ASCII to binary conversion

# Graphical user interface Description automatically generated

# Wap to implement linear search.

LXI H,8000H

MOV C, M

INX H

MOV B, M

MOV A, B

INX H

CMP M

JZ FOUND

DCR C

JNZ LOOP

LXI H, FFFF

SHLD 9000H

JMP DONE

SHLD 9000H

HLT

# Wap to implement binary search

# .model small

# .data

# array db 01h,02h,03h,05h,0ah,0eh,10h,18h,33h,0ffh

# .code

# .startup

# start:

# mov al,18h

# mov dl,00h

# mov dh,09h

# mov bx,0000h

# mov ah,01h

# bsearch:

# mov bl,dl

# add bl,dh

# shr bx,1

# .if array[bx]==al

# mov ah,00h

# jmp stop

# .elseif al>array[bx]

# mov dl,bl

# add dl,01h

# .else

# mov dh,bl

# sub dh,01h

# .endif

# cmp dl,dh

# jle bsearch

# stop:

# .if ah==01h

# mov bx,0ffffh

# .endif

# .exit

# end