NAME: **GINI CHACKO**

SEMESTER: **IV**

CLASS: **SE COMPS B**

BATCH: **B**

ROLL: **8942**

TOPIC: **MP EXPERIMENT 1 :**

8 bit ADDITION

8 bit SUBTRACTION

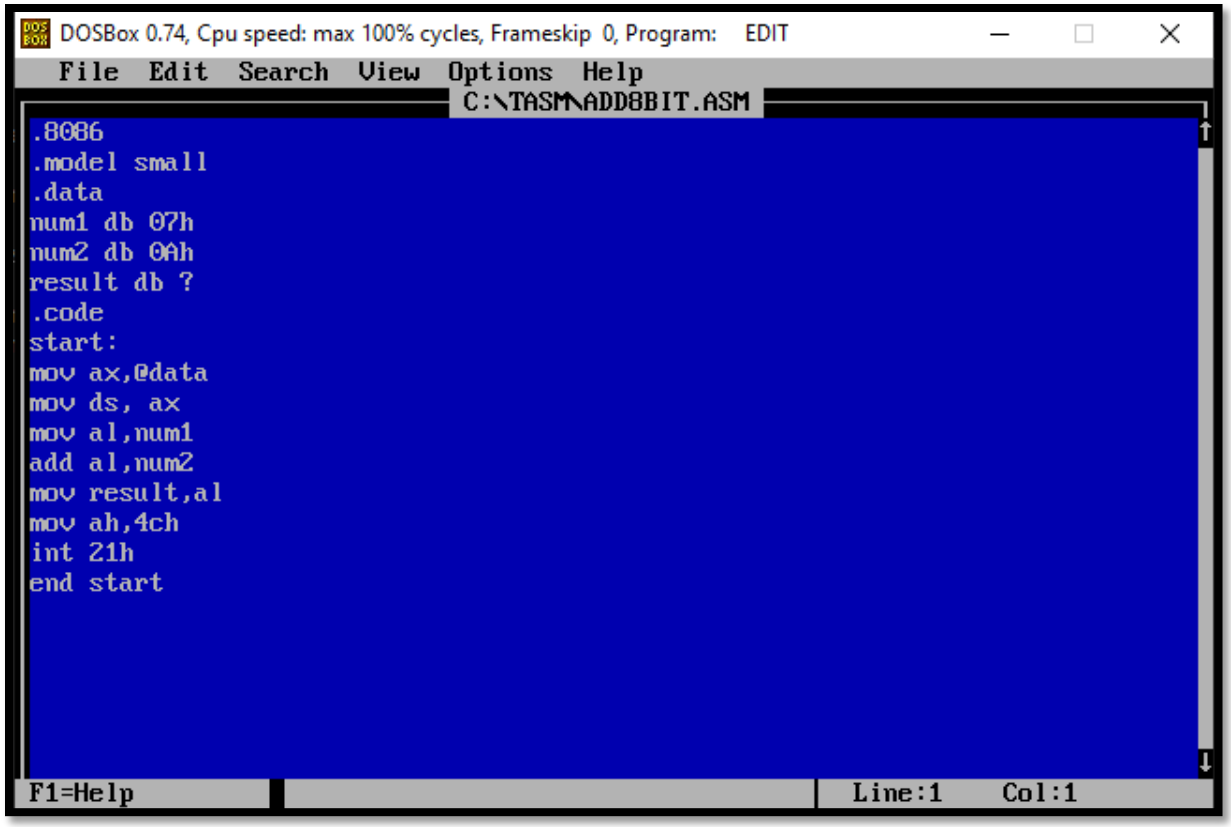
8 bit MULTIPLICATION

8 bit DIVISION

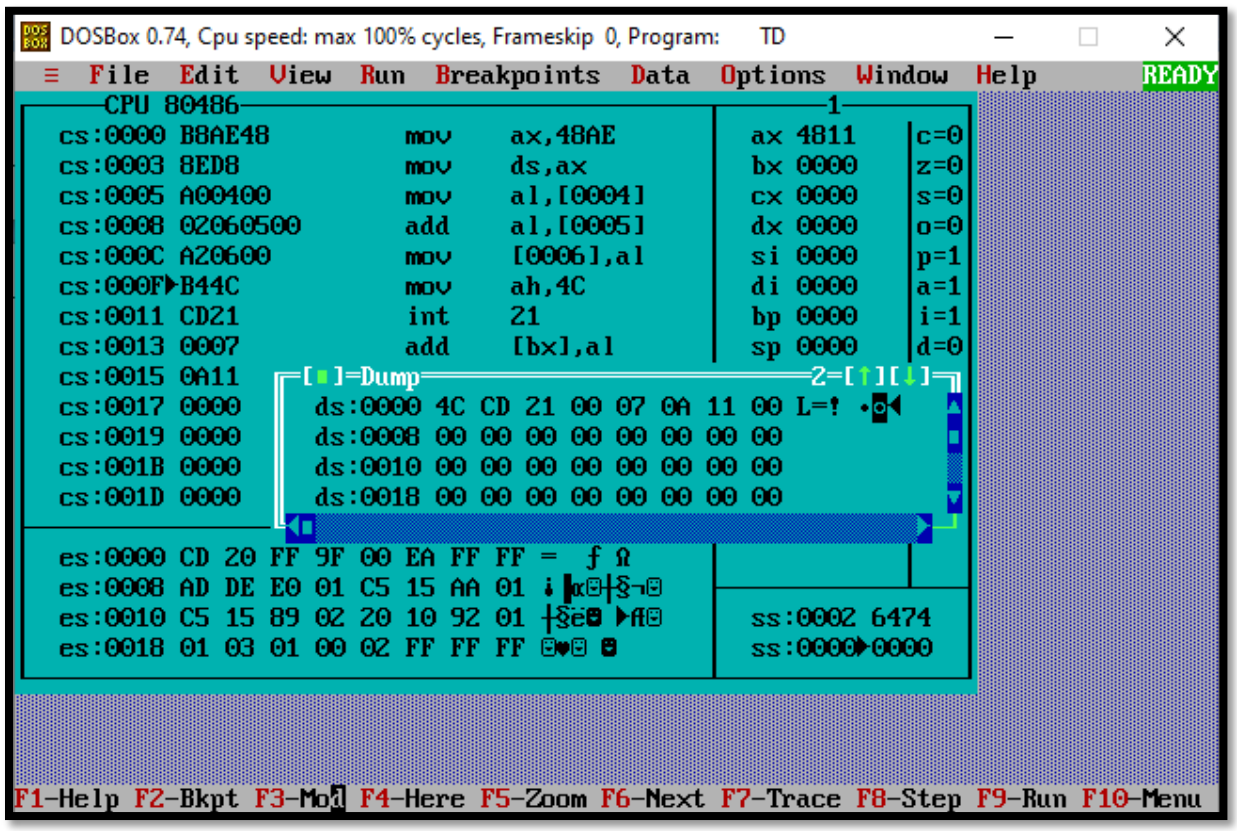
❖ 8 bit ADDITION :

CODE:

```
.8086
.model small
.data
num1 db 07h
num2 db 0Ah
result db ?
.code
start:
mov ax,@data
mov ds, ax
mov al,num1
add al,num2
mov result,al
mov ah,4ch
int 21h
end start
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\TASM\ADD8BIT.ASM
.8086
.model small
.data
num1 db 07h
num2 db 0Ah
result db ?
.code
start:
mov ax,@data
mov ds, ax
mov al,num1
add al,num2
mov result,al
mov ah,4ch
int 21h
end start
F1=Help
Line:1 Col:1

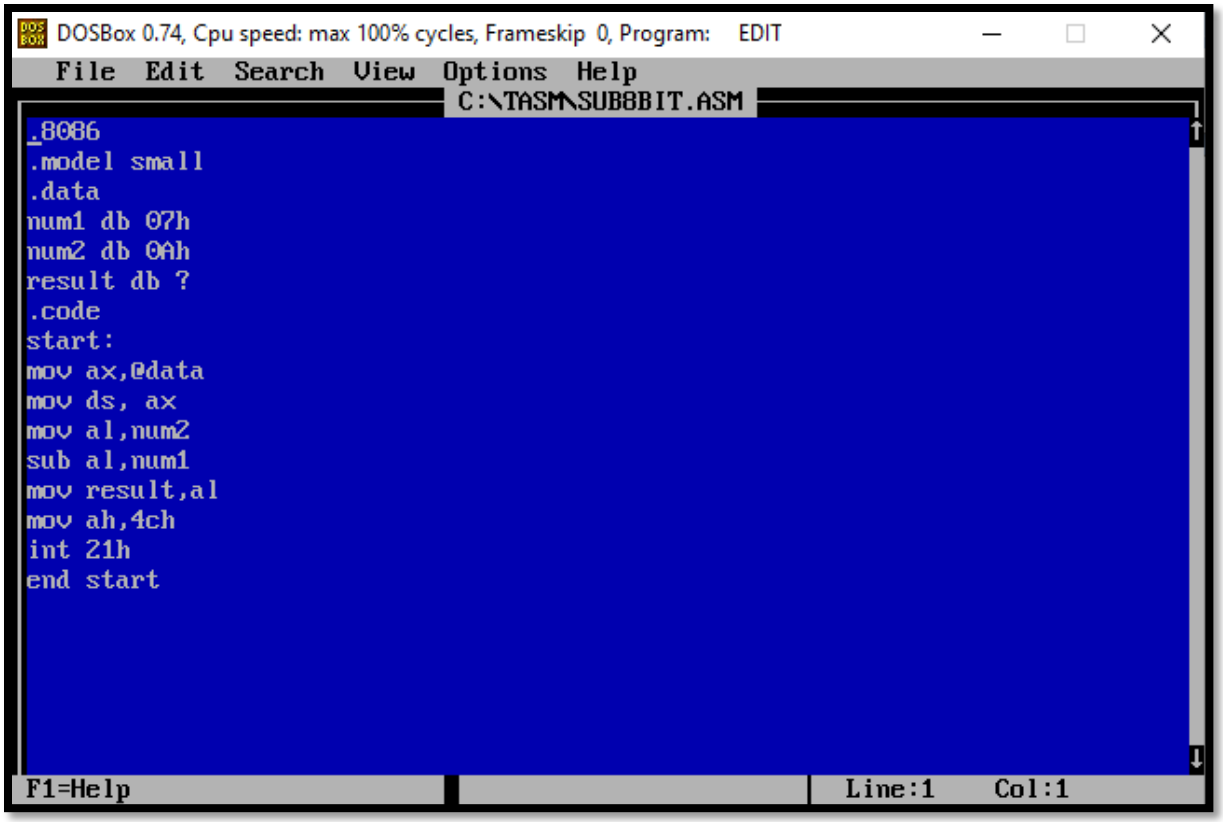
OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:0000 B8AE48 mov ax,48AE
cs:0003 8ED8 mov ds,ax
cs:0005 A00400 mov al,[0004]
cs:0008 02060500 add al,[0005]
cs:000C A20600 mov [0006],al
cs:000F B44C mov ah,4C
cs:0011 CD21 int 21
cs:0013 0007 add [bx],al
cs:0015 0A11
cs:0017 0000
cs:0019 0000
cs:001B 0000
cs:001D 0000
ax 4811 c=0
bx 0000 z=0
cx 0000 s=0
dx 0000 o=0
si 0000 p=1
di 0000 a=1
bp 0000 i=1
sp 0000 d=0
Dump
ds:0000 4C CD 21 00 07 0A 11 00
ds:0008 00 00 00 00 00 00 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
es:0000 CD 20 FF 9F 00 EA FF FF
es:0008 AD DE E0 01 C5 15 AA 01
es:0010 C5 15 89 02 20 10 92 01
es:0018 01 03 01 00 02 FF FF FF
ss:0002 6474
ss:0000 0000
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

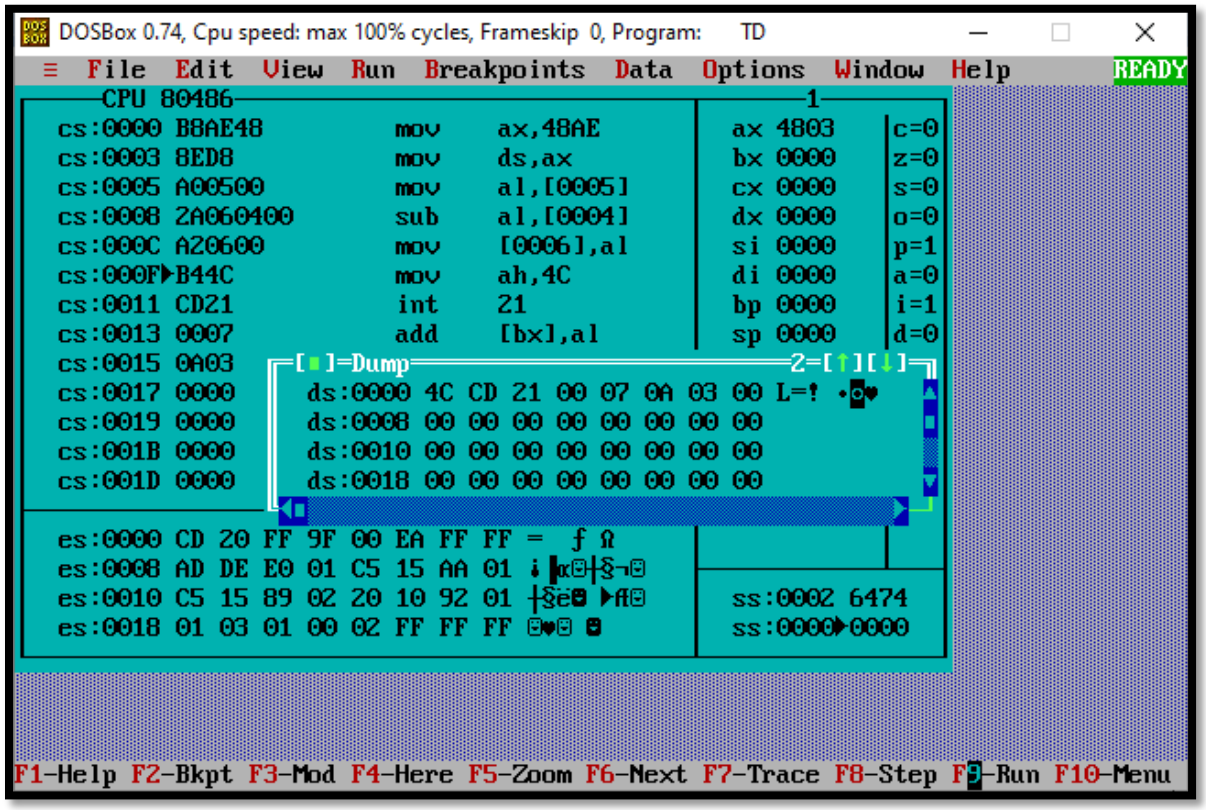
❖ 8 bit SUBTRACTION :

CODE:

```
.8086
.model small
.data
num1 db 07h
num2 db 0Ah
result db ?
.code
start:
mov ax,@data
mov ds, ax
mov al,num2
sub al,num1
mov result,al
mov ah,4ch
int 21h
end start
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\TASM\SUB8BIT.ASM
.8086
.model small
.data
num1 db 07h
num2 db 0Ah
result db ?
.code
start:
mov ax,@data
mov ds, ax
mov al,num2
sub al,num1
mov result,al
mov ah,4ch
int 21h
end start
F1=Help
Line:1 Col:1

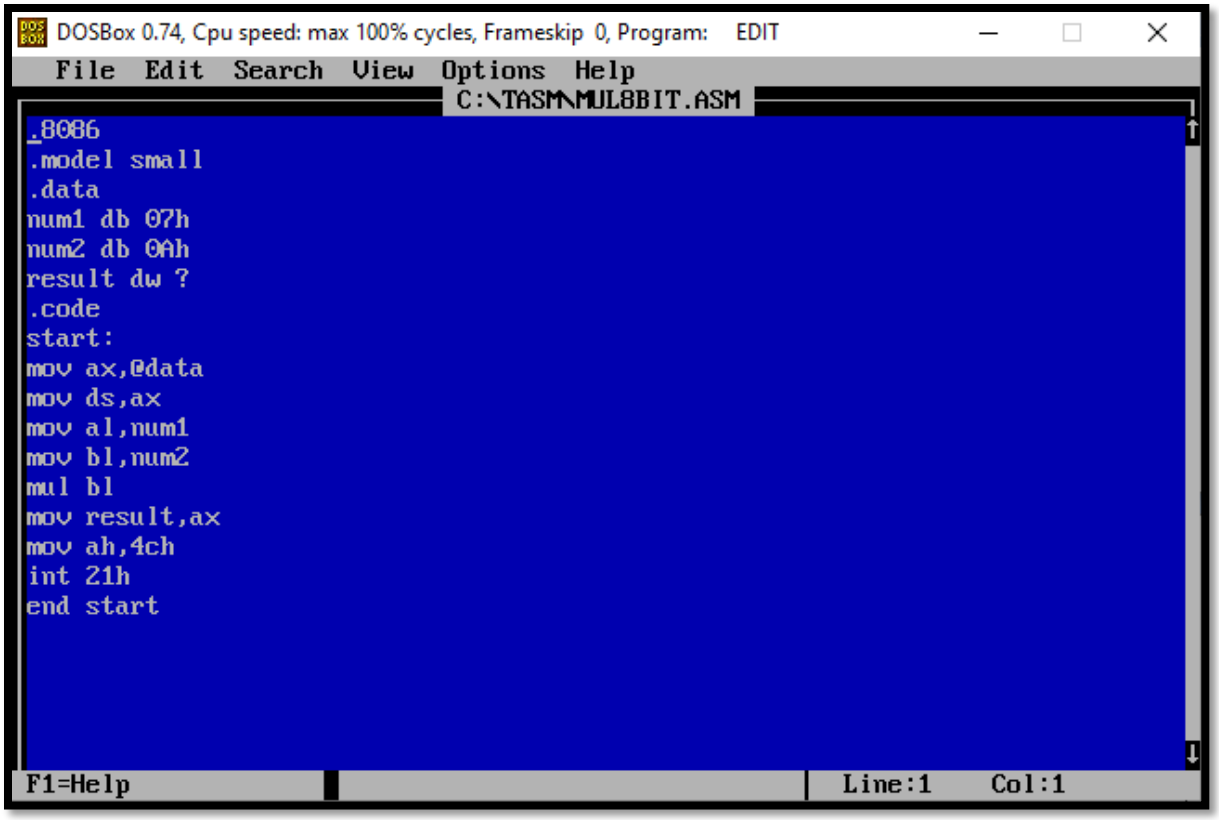
OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help READY
CPU 80486
cs:0000 B8AE48 mov ax,48AE
cs:0003 8ED8 mov ds,ax
cs:0005 A00500 mov al,[0005]
cs:0008 2A060400 sub al,[0004]
cs:000C A20600 mov [0006],al
cs:000F B44C mov ah,4C
cs:0011 CD21 int 21
cs:0013 0007 add [bx],al
cs:0015 0A03
cs:0017 0000
cs:0019 0000
cs:001B 0000
cs:001D 0000
Dump
ds:0000 4C CD 21 00 07 0A 03 00 L=!
ds:0008 00 00 00 00 00 00 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
es:0000 CD 20 FF 9F 00 EA FF FF
es:0008 AD DE E0 01 C5 15 AA 01
es:0010 C5 15 89 02 20 10 92 01
es:0018 01 03 01 00 02 FF FF FF
ax 4803 c=0
bx 0000 z=0
cx 0000 s=0
dx 0000 o=0
si 0000 p=1
di 0000 a=0
bp 0000 i=1
sp 0000 d=0
ss:0002 6474
ss:0000 0000
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

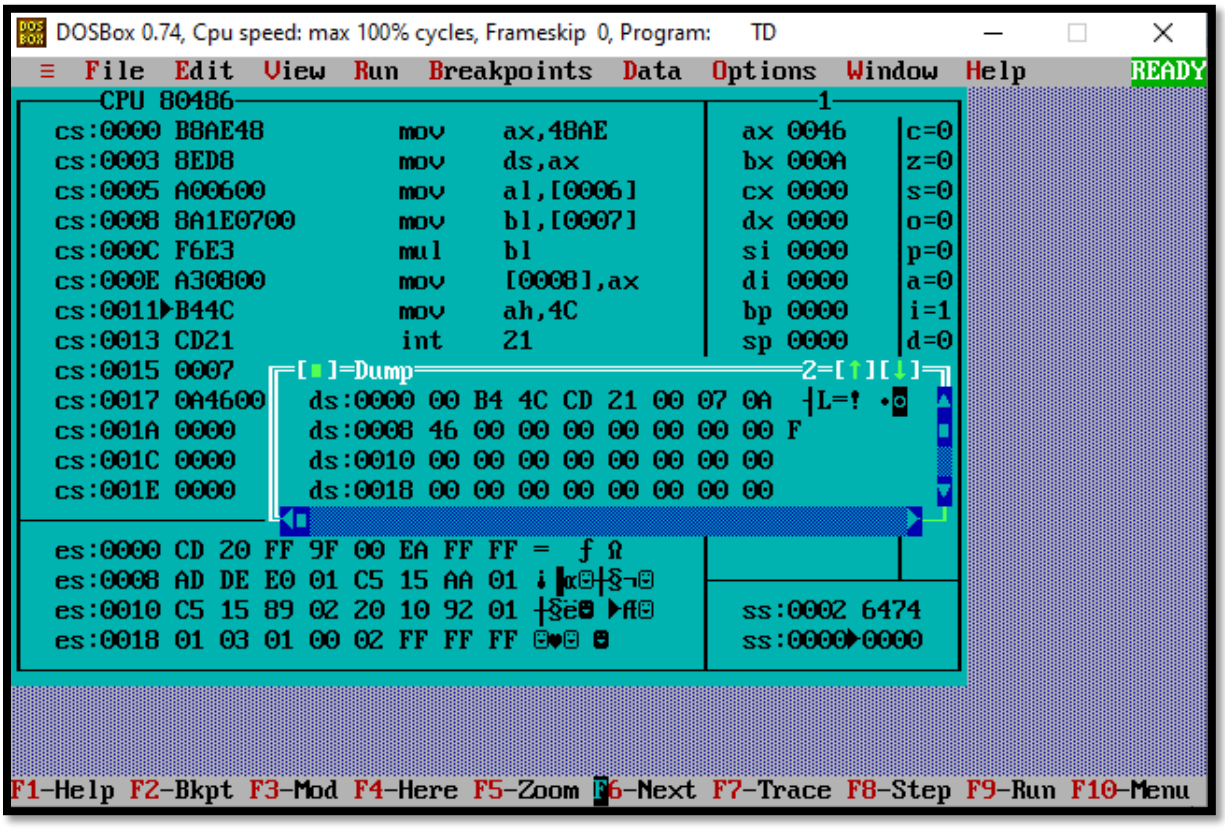
❖ 8 bit MULTIPLICATION :

CODE:

```
.8086
.model small
.data
num1 db 07h
num2 db 0Ah
result dw ?
.code
start:
mov ax,@data
mov ds,ax
mov al,num1
mov bl,num2
mul bl
mov result,ax
mov ah,4ch
int 21h
end start
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\TASM\MUL8BIT.ASM
.8086
.model small
.data
num1 db 07h
num2 db 0Ah
result dw ?
.code
start:
mov ax,@data
mov ds,ax
mov al,num1
mov bl,num2
mul bl
mov result,ax
mov ah,4ch
int 21h
end start
F1=Help
Line:1 Col:1

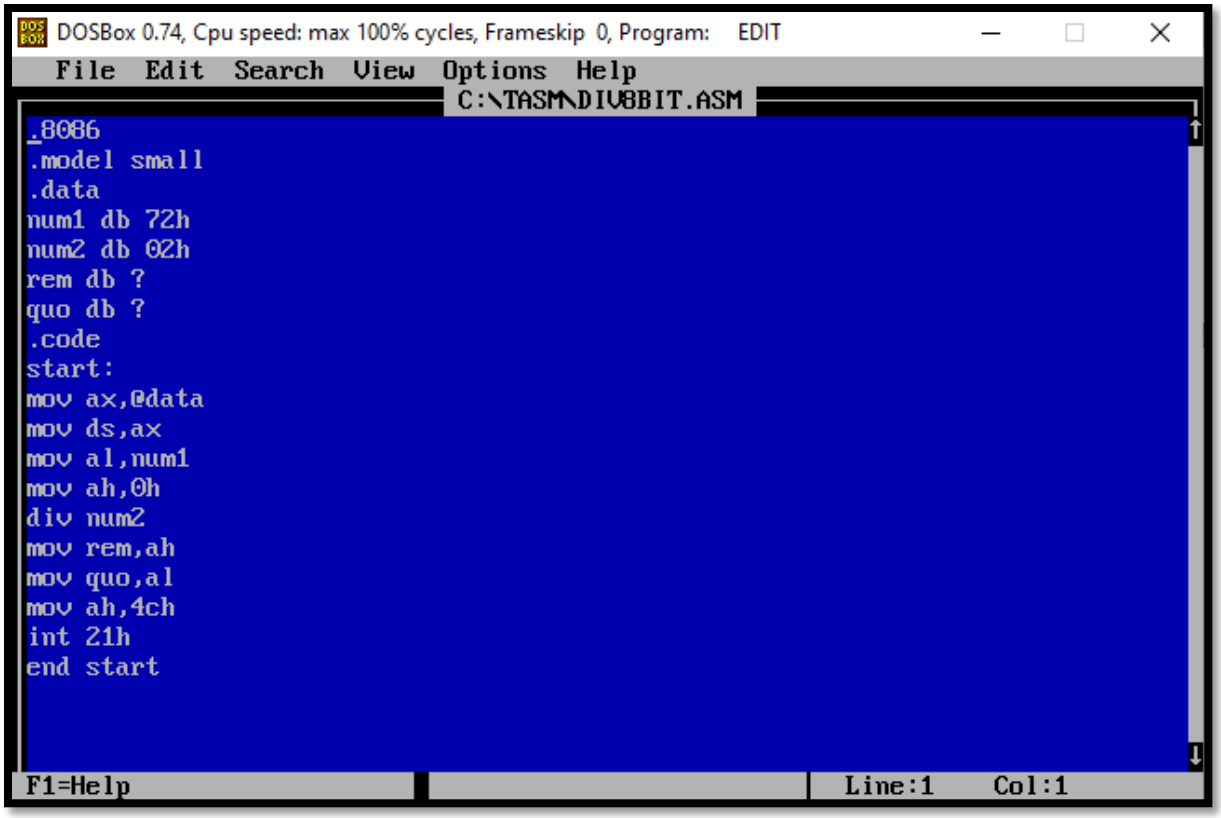
OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help READY
CPU 80486
cs:0000 B8AE48 mov ax,48AE
cs:0003 8ED8 mov ds,ax
cs:0005 A00600 mov al,[0006]
cs:0008 8A1E0700 mov bl,[0007]
cs:000C F6E3 mul bl
cs:000E A30800 mov [0008],ax
cs:0011 B44C mov ah,4C
cs:0013 CD21 int 21
cs:0015 0007
cs:0017 0A4600
cs:001A 0000
cs:001C 0000
cs:001E 0000
[■]=Dump
ds:0000 00 B4 4C CD 21 00 07 0A
ds:0008 46 00 00 00 00 00 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
es:0000 CD 20 FF 9F 00 EA FF FF
es:0008 AD DE E0 01 C5 15 AA 01
es:0010 C5 15 89 02 20 10 92 01
es:0018 01 03 01 00 02 FF FF FF
ax 0046
bx 000A
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0000
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
ss:0002 6474
ss:0000 0000
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

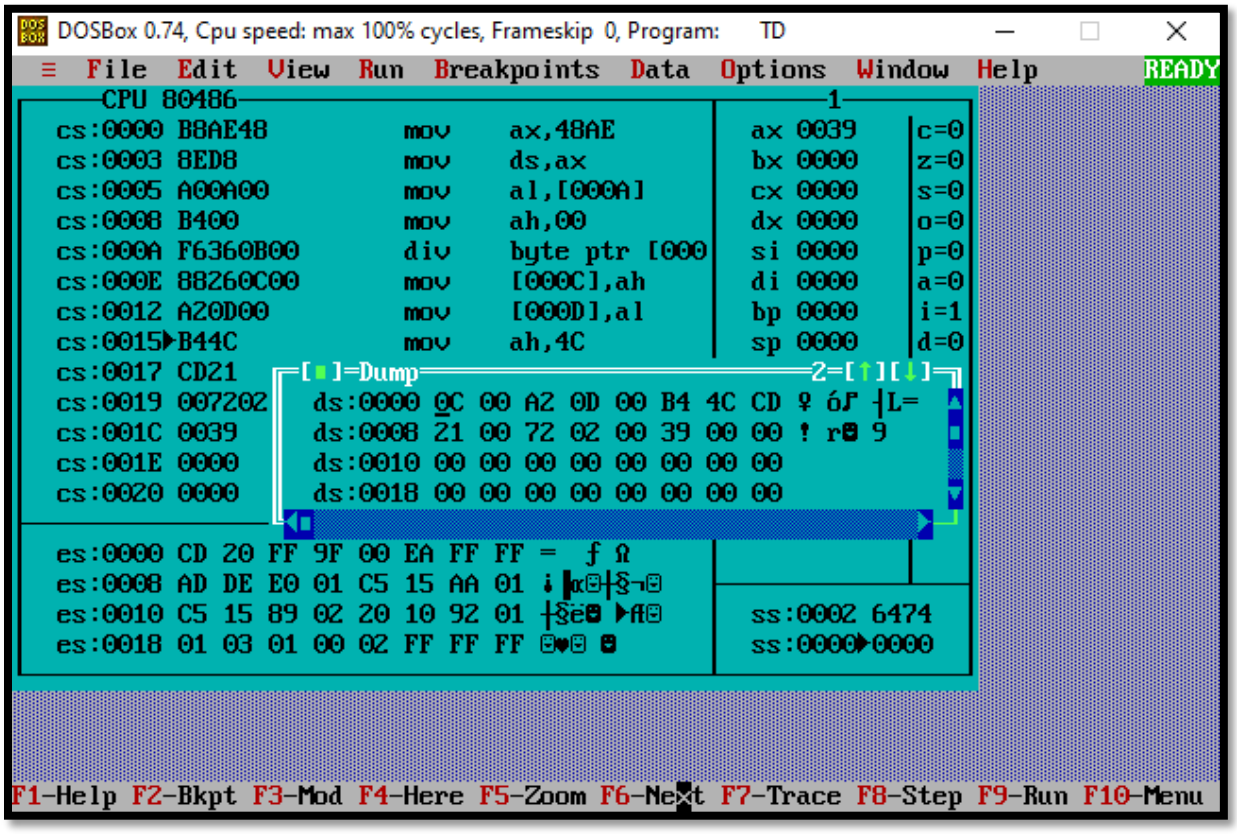
❖ 8 bit DIVISION :

CODE:

```
.8086
.model small
.data
num1 db 72h
num2 db 02h
rem db ?
quo db ?
.code
start:
mov ax,@data
mov ds,ax
mov al,num1
mov ah,0h
div num2
mov rem,ah
mov quo,al
mov ah,4ch
int 21h
end start
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\TASM\DIV8BIT.ASM
.8086
.model small
.data
num1 db 72h
num2 db 02h
rem db ?
quo db ?
.code
start:
mov ax,@data
mov ds,ax
mov al,num1
mov ah,0h
div num2
mov rem,ah
mov quo,al
mov ah,4ch
int 21h
end start
F1=Help
Line:1 Col:1

OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:0000 B8AE48 mov ax,48AE
cs:0003 8ED8 mov ds,ax
cs:0005 A00A00 mov al,[000A]
cs:0008 B400 mov ah,00
cs:000A F6360B00 div byte ptr [000
cs:000E 88260C00 mov [000C],ah
cs:0012 A20D00 mov [000D],al
cs:0015 B44C mov ah,4C
cs:0017 CD21
cs:0019 007202
cs:001C 0039
cs:001E 0000
cs:0020 0000
ax 0039
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0000
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
[■]=Dump
2=[↑][↓]
ds:0000 0C 00 A2 0D 00 B4 4C CD
ds:0008 21 00 72 02 00 39 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
es:0000 CD 20 FF 9F 00 EA FF FF
es:0008 AD DE E0 01 C5 15 AA 01
es:0010 C5 15 89 02 20 10 92 01
es:0018 01 03 01 00 02 FF FF FF
ss:0002 6474
ss:0000 0000
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

POSTLAB QUESTIONS:

1.] Write the 8086 CPU architecture.

ANS:

- 8086 Microprocessor is an enhanced version of 8085 Microprocessor that was designed by Intel in 1976
- It is a 16-bit microprocessor having 20 address lines and 16 data lines that provides up to 1MB storage.
- It consists of powerful instruction set, which provides operations like multiplication and division easily.
- It supports two modes of operation, i.e maximum mode and minimum mode.
- Maximum mode is suitable for system having multiple processors and minimum mode is suitables for system having a single processor.

Features of 8086

1) The 8086, has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
2) It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
3) It is available in 3 versions based on the frequency of operation.
4) It uses 2 stages of pipelining, i.e fetch stage and and stores them in the queue.
5) Execute stage executes these instructions.
6) It has 256 vectored interrupts and 29,000 transistors.

2.] Give the syntax to write assembly instruction format.

ANS: 1) The assembly text is usually divided into fields, separated by spaces and tabs. A format for a typical line from assembly language program can be given as

Label : Mnemonic Operand1, Operand2 ; comment

2) The first field, which is optional, is the label field, used to specify symbolic labels. A label is an identifier that is assigned to the address of the first byte of the instruction in which it appears. The label provides a symbolic name that can be used in branch instructions to branch to the instruction.
3) The second field is mnemonic, which is compulsory. All instructions must contain a mnemonic.
4) The third and following fields are operands. The presence of the operands depends on the instruction. Some instructions have no operands, some have one, and some have two.
5) The last field is a comment field. It begins with a delimiter such as the semicolon and continues to the end of the line.
6) The comments are for our benefits, they tell us what the program is trying to accomplish.