

Experiment No 1: 8-bit Arithmetic Operations

Date: 17-08-2020

NAME: Harshini S

REG.NO: 185001058

1. AIM:

Program for adding 2, 8-bit numbers.

PROCEDURE FOR EXECUTING MASM:

- Run Dosbox and mount the masm folder to a drive in dosbox
- Goto the mounted drive ("Drive:")
- Save the 8086 program with extension .asm in the same folder using command "edit" in Dosbox.
- Next, assemble it using the command "masm filename.asm"
- Link the file using the command "link filename.obj;"
- Debug the file to execute and analyse the memory contents, "debug filename.exe".
- Now use command "u" to display the unassembled code.
- Use command "d segment:offset" to see the content of memory locations starting from segment:offset address
- Execute using the command "g" and check the outputs by repeating the previous step.
- Use command "e segment:offset" to edit the variables.
- Command "q" to exit from debug and command "exit" from command prompt to close dosbox.

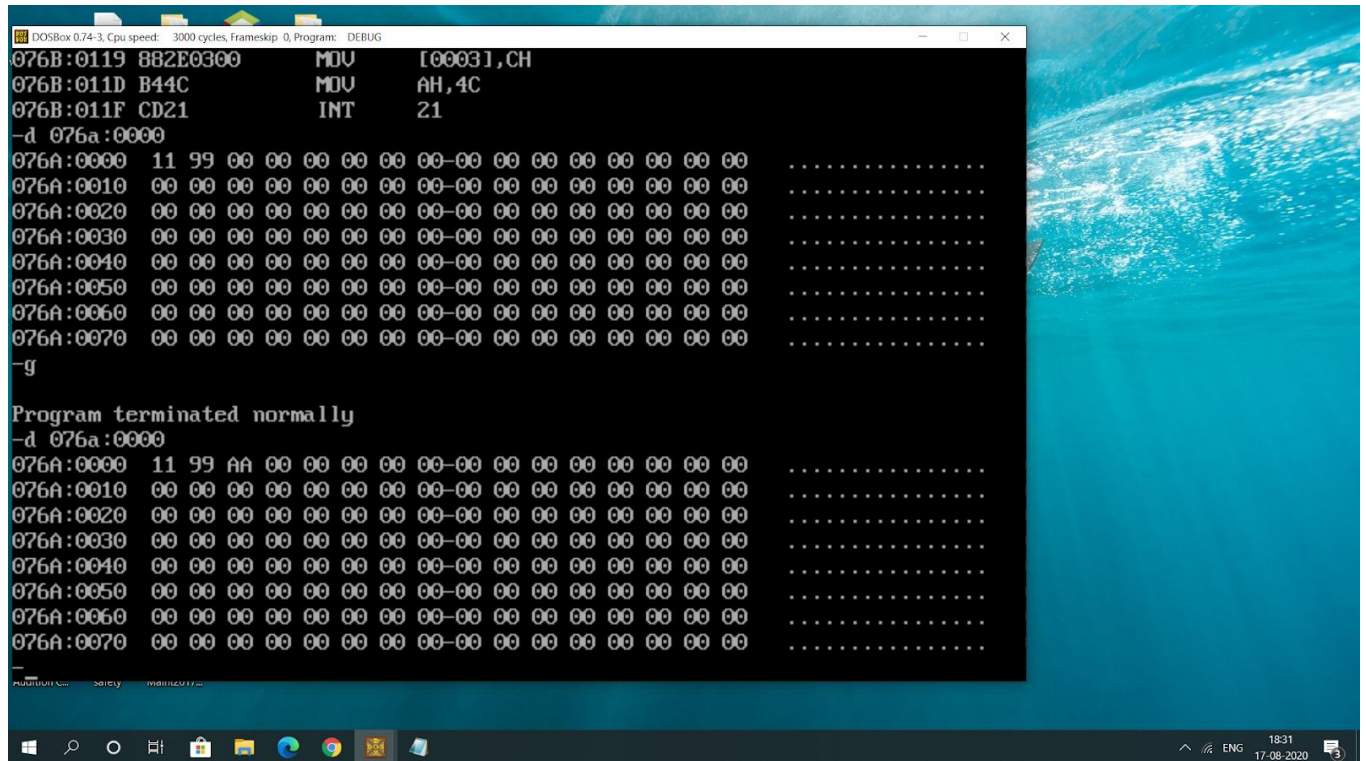
ALGORITHM:

- Initialize the data segment
- Move data segment address to ds
- Load operand-1 to ah and operand-2 to bh
- Load 00h to ch register for carry
- Add ah and bh
- If there is no carry being generated, goto here segment else, increment ch by 1
- In here segment,
 - Load ah to result
 - Load ch to carry
 - Terminate the program

PROGRAM:

PROGRAM	COMMENTS
Start: mov ax,data mov ds,ax mov ah,opr1 mov bh,opr2 mov ch,00h add ah,bh jnc here inc ch	Transferring address of data segment to ds Value of opr1 is loaded to ah Value of opr2 is loaded to bh Initializing the value of ch ah=ah+bh Jump to “here” segment if no carry is generated Increments ch by 1
Here: mov result,ah mov carry,ch mov ah,4ch int 21h code ends	 Load register value of ah to result Load ch value to carry Termination of execution Ending the segment with the segment name

(ah=11 ; bh=99)



RESULT:

The addition of 2, 8-bit numbers is thus shown.

2. AIM:

Program for subtracting 2, 8-bit numbers.

PROCEDURE FOR EXECUTING MASM:

- Run Dosbox and mount the masm folder to a drive in dosbox
- Goto the mounted drive ("Drive:")
- Save the 8086 program with extension .asm in the same folder using command "edit" in Dosbox.
- Next, assemble it using the command "masm filename.asm"
- Link the file using the command "link filename.obj;"
- Debug the file to execute and analyse the memory contents, "debug filename.exe".
- Now use command "u" to display the unassembled code.
- Use command "d segment:offset" to see the content of memory locations starting from segment:offset address
- Execute using the command "g" and check the outputs by repeating the previous step.
- Use command "e segment:offset" to edit the variables.
- Command "q" to exit from debug and command "exit" from command prompt to close dosbox.

ALGORITHM:

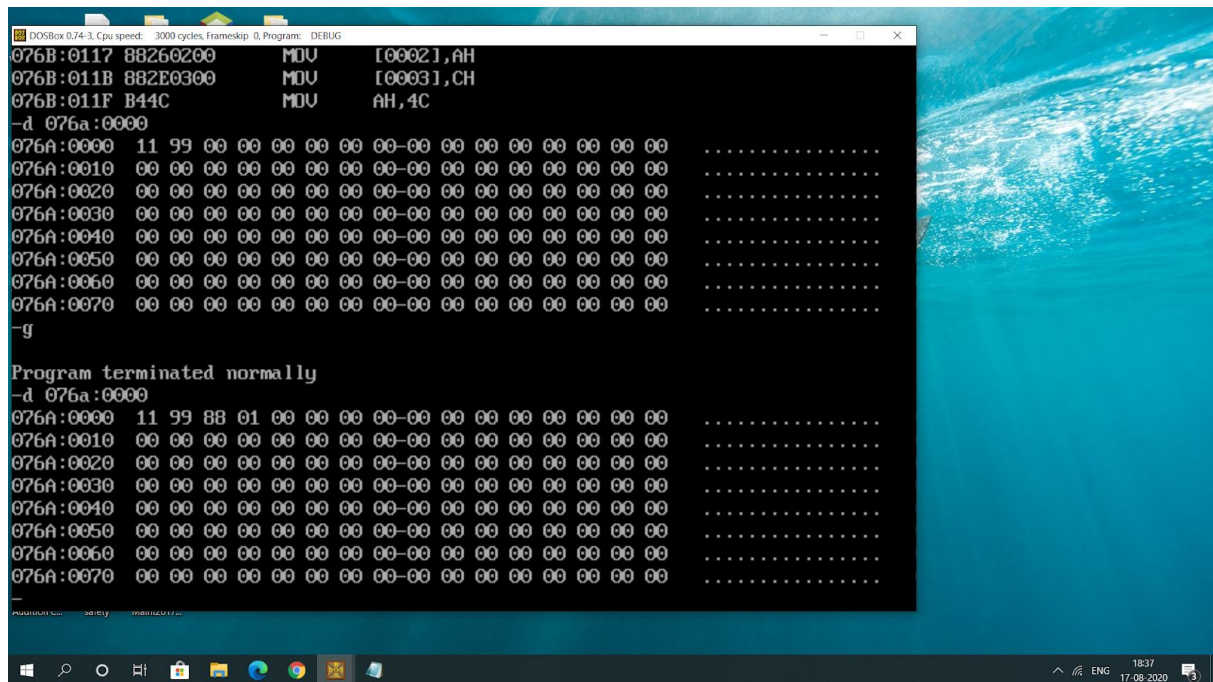
- Initialize the data segment
- Move data segment address to ds
- Load operand-1 to ah and operand-2 to bh
- Load 00h to ch register
- Subtract ah and bh
- If ah is greater than bh, goto here segment else, increment ch by 1 and find the 2's complement of ah
- In here segment,
 - Load ah to result
 - Load ch to carry
 - Terminate the program

PROGRAM:

PROGRAM	COMMENTS
Start: mov ax,data mov ds,ax mov ah,opr1 mov bh,opr2 mov ch,00h sub ah,bh jnc here inc ch neg ah	Transferring address of data segment to ds Value of opr1 is loaded to ah Value of opr2 is loaded to bh Initializing the value of ch ah=ah-bh Jump to “here” segment if ah>bh Increments ch by 1 2’s complement of ah
Here: mov result,ah mov carry,ch mov ah,4ch int 21h code ends	Load register value of ah to result Load ch value to carry Termination of execution Ending the segment with the segment name

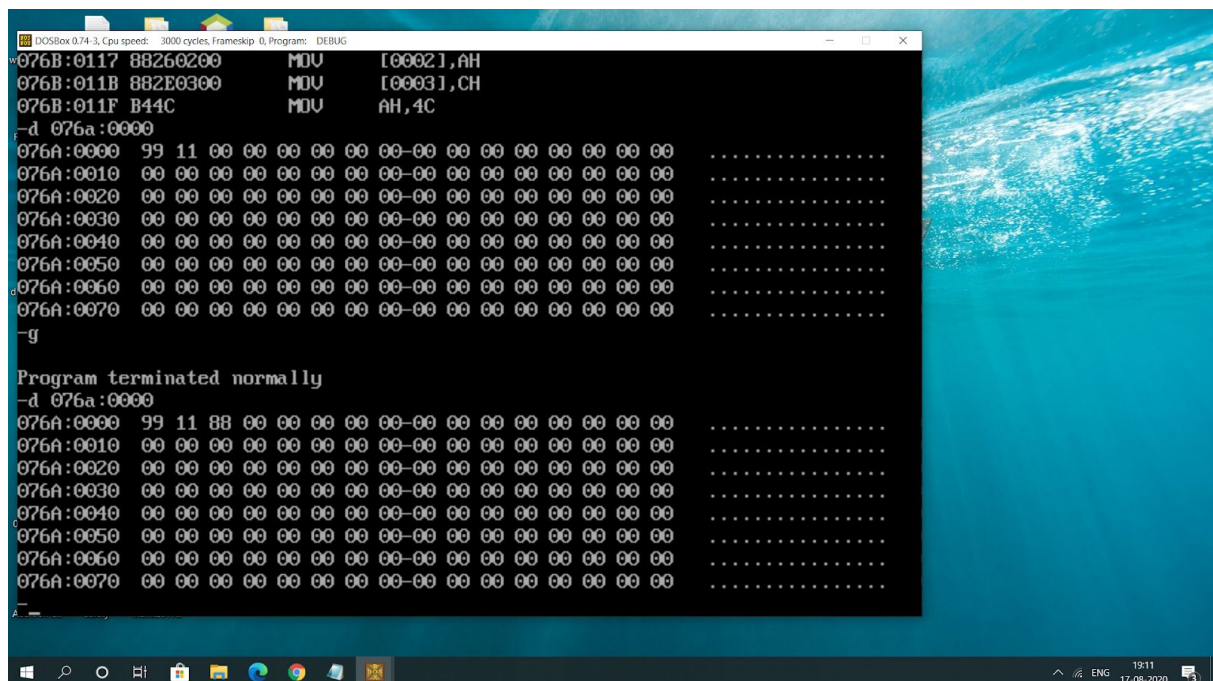
SAMPLE INPUT/OUTPUT

ah=11; bh=99 (ah<bh)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG
076B:0117 88260200      MOV     [0002],AH
076B:011B 882E0300      MOV     [0003],CH
076B:011F B44C      MOV     AH,4C
-d 076a:0000
076A:0000 11 99 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
-g
Program terminated normally
-d 076a:0000
076A:0000 11 99 88 01 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

ah=99 bh=11 (ah>bh)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG
076B:0117 88260200      MOV     [0002],AH
076B:011B 882E0300      MOV     [0003],CH
076B:011F B44C      MOV     AH,4C
-d 076a:0000
076A:0000 99 11 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
-g
Program terminated normally
-d 076a:0000
076A:0000 99 11 88 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

RESULT:

The subtraction of 2, 8-bit numbers is thus shown.

3. AIM:

Program for multiplication of 2, 8-bit numbers.

PROCEDURE FOR EXECUTING MASM:

- Run Dosbox and mount the masm folder to a drive in dosbox
- Goto the mounted drive ("Drive:")
- Save the 8086 program with extension .asm in the same folder using command "edit" in Dosbox.
- Next, assemble it using the command "masm filename.asm"
- Link the file using the command "link filename.obj;"
- Debug the file to execute and analyse the memory contents, "debug filename.exe".
- Now use command "u" to display the unassembled code.
- Use command "d segment:offset" to see the content of memory locations starting from segment:offset address
- Execute using the command "g" and check the outputs by repeating the previous step.
- Use command "e segment:offset" to edit the variables.
- Command "q" to exit from debug and command "exit" from command prompt to close dosbox.

ALGORITHM:

- Initialize the data segment
- Move data segment address to ds
- Load operand-1 to al and operand-2 to bl
- Multiply bl ($ax = al \times bl$)
- Load ax to result
- Terminate the program

PROGRAM:

PROGRAM	COMMENTS
Start:	
mov ax,data mov ds,ax	Transferring address of data segment to ds
mov al,opr1	Value of opr1 is loaded to al
mov bl,opr2	Value of opr2 is loaded to bl
mul bl	ax=al x bl
mov result,ax	Load register value of ax to result
mov ah,4ch int 21h	Termination of execution
code ends	Ending the segment with the segment name

SAMPLE INPUT/OUTPUT (al=11 ; bl=99)

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, frameskip: 0, Program: DEBUG
076B:011B B0FF      MOV     AL,FF
076B:011D 7701      JA      0120
076B:011F 40       INC     AX
-d 076a:0000
076A:0000 11 99 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
-g
Program terminated normally
-d 076a:0000
076A:0000 11 99 29 0A 00 00 00 00 00 00 00 00 00 00 00 00 ..).....
076A:0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

RESULT:

The multiplication of 2, 8-bit numbers is thus shown.

4. AIM:

Program for division of 2, 8-bit numbers.

PROCEDURE FOR EXECUTING MASM:

- Run Dosbox and mount the masm folder to a drive in dosbox
- Goto the mounted drive ("Drive:")
- Save the 8086 program with extension .asm in the same folder using command "edit" in Dosbox.
- Next, assemble it using the command "masm filename.asm"
- Link the file using the command "link filename.obj;"
- Debug the file to execute and analyse the memory contents, "debug filename.exe".
- Now use command "u" to display the unassembled code.
- Use command "d segment:offset" to see the content of memory locations starting from segment:offset address
- Execute using the command "g" and check the outputs by repeating the previous step.
- Use command "e segment:offset" to edit the variables.
- Command "q" to exit from debug and command "exit" from command prompt to close dosbox.

ALGORITHM:

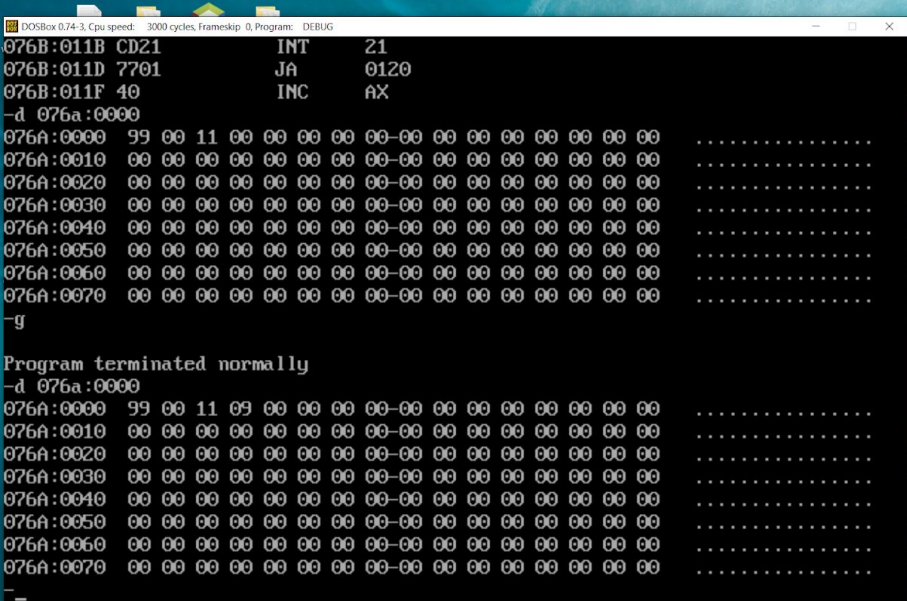
- Initialize the data segment
- Move data segment address to ds
- Load ah with 00
- Load operand-1 to ax and operand-2 to bl
- Divide bl ($al = ax / bl$; remainder in ah)
- Load al to result
- Load ah to rem (remainder)
- Terminate the program

PROGRAM:

PROGRAM	COMMENTS
Start: mov ax,data mov ds,ax mov ah,00 mov ax,opr1 mov bl,opr2 div bl mov result,al mov rem,ah mov ah,4ch int 21h code ends	Transferring address of data segment to ds Register ah is loaded with 00 Value of opr1 is loaded to ax Value of opr2 is loaded to bl $al = ax / bl$ Load register value of al to result Load register value of ah to rem Termination of execution Ending the segment with the segment name

SAMPLE INPUT/OUTPUT

(ax=99 ; bl=11)

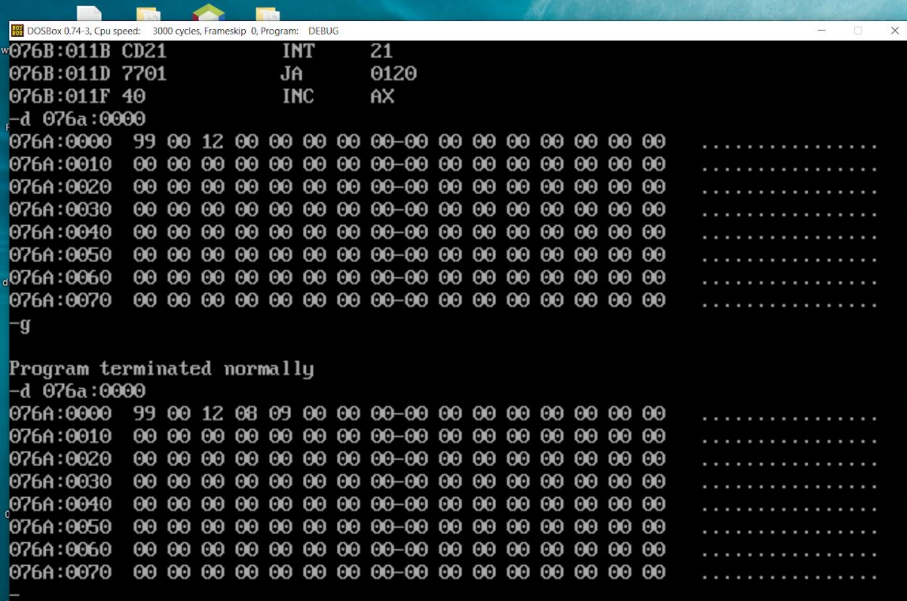


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG

```
076B:011B CD21      INT     21
076B:011D 7701      JA      0120
076B:011F 40        INC     AX
-d 076a:0000
076A:0000 99 00 11 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
-g
Program terminated normally
-d 076a:0000
076A:0000 99 00 11 09 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

Windows taskbar shows: 18:46, 17-08-2020

(ax=99 ; bl=12)



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG

```
076B:011B CD21      INT     21
076B:011D 7701      JA      0120
076B:011F 40        INC     AX
-d 076a:0000
076A:0000 99 00 12 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
-g
Program terminated normally
-d 076a:0000
076A:0000 99 00 12 08 09 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

Windows taskbar shows: 19:14, 17-08-2020

RESULT:

The division of 2, 8-bit numbers is thus shown.