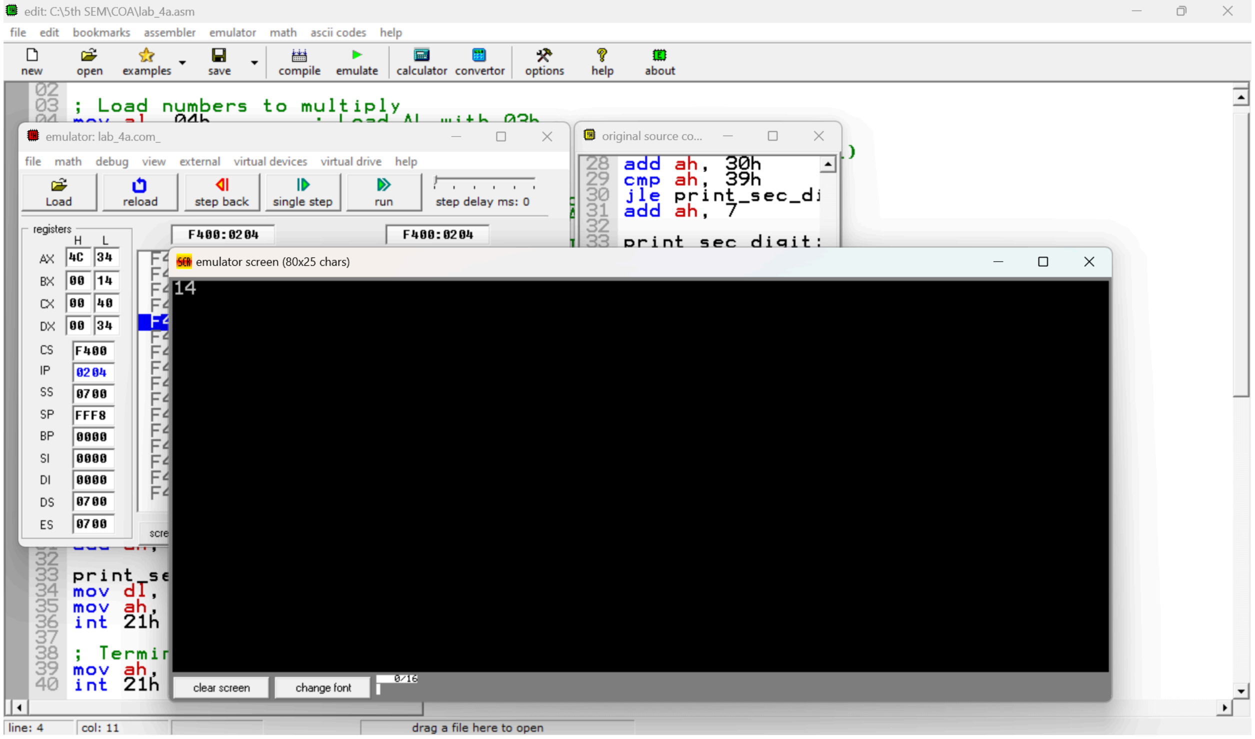
Write an assembly language program to perform multiplication of 8-bit data.

```
org 100h           ; Set starting address

; Load numbers to multiply
mov al, 12h        ; Load AL with 03h
mov bl, 05h        ; Load BL with 04h
mul bl             ; Multiply AL by BL, result in AX (AL * BL)

; Move the result to BL for later use
mov bl, al         ; Move result (AL) to BL
mov ah, al         ; Move AL to AH for ASCII conversion

; Convert upper nibble of result to ASCII
and ah, 0F0h       ; Mask lower nibble, keep upper
shr ah, 4          ; Shift upper nibble to lower position
add ah, 30h        ; Convert to ASCII '0'-'9'
cmp ah, 39h        ; Compare with '9'
jle print_first_digit ; If less or equal to '9', skip next step
add ah, 7          ; Convert to ASCII 'A'-'F' (if necessary)

print_first_digit:
mov dl, ah         ; Move first digit to DL
mov ah, 02h        ; Prepare for output (DOS interrupt 21h, function 02h)
int 21h            ; Print first digit

; Convert lower nibble of result to ASCII
mov ah, bl         ; Move result (BL) back to AH
and ah, 0Fh        ; Mask upper nibble, keep lower
add ah, 30h        ; Convert to ASCII '0'-'9'
cmp ah, 39h        ; Compare with '9'
jle print_sec_digit ; If less or equal to '9', skip next step
add ah, 7          ; Convert to ASCII 'A'-'F' (if necessary)

print_sec_digit:
mov dl, ah         ; Move second digit to DL
mov ah, 02h        ; Prepare for output
int 21h            ; Print second digit

; Terminate the program
mov ah, 4Ch        ; Prepare for program termination
int 21h            ; Terminate program
```

OUTPUT:


edit: C:\5th SEM\COA\lab_4a.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: lab_4a.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
emulator screen (80x25 chars)
clear screen change font
line: 4 col: 11 drag a file here to open

```
; Load numbers to multiply
```

```
add ah, 30h
cmp ah, 39h
jle print_sec_di
add ah, 7

print_sec_digit:
```

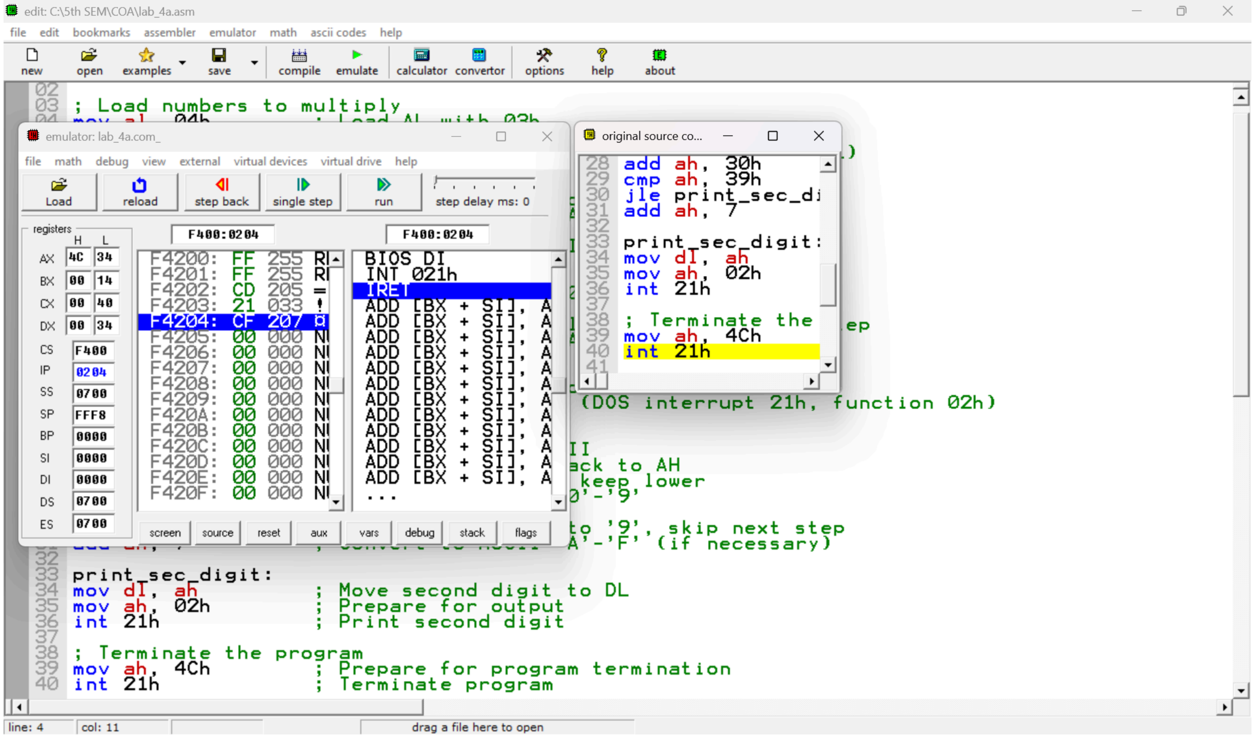
registers

	H	L
AX	4C	34
BX	00	14
CX	00	40
DX	00	34
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204 F400:0204

```
14
```

```
print_se
mov dl,
mov ah,
int 21h

; Termin
mov ah,
int 21h
```


edit: C:\5th SEM\COA\lab_4a.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: lab_4a.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
screen source reset aux vars debug stack flags
line: 4 col: 11 drag a file here to open

```
; Load numbers to multiply
```

registers

	H	L
AX	4C	34
BX	00	14
CX	00	40
DX	00	34
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204 F400:0204

```
F4200:  FF  255  RI
F4201:  FF  255  RI
F4202:  CD  205  =
F4203:  21  033  !
F4204:  CF  207  ¤
F4205:  00  000  NI
F4206:  00  000  NI
F4207:  00  000  NI
F4208:  00  000  NI
F4209:  00  000  NI
F420A:  00  000  NI
F420B:  00  000  NI
F420C:  00  000  NI
F420D:  00  000  NI
F420E:  00  000  NI
F420F:  00  000  NI
```

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
...
```

```
add ah, 30h
cmp ah, 39h
jle print_sec_di
add ah, 7

print_sec_digit:
mov dl, ah
mov ah, 02h
int 21h

; Terminate the
mov ah, 4Ch
int 21h
```

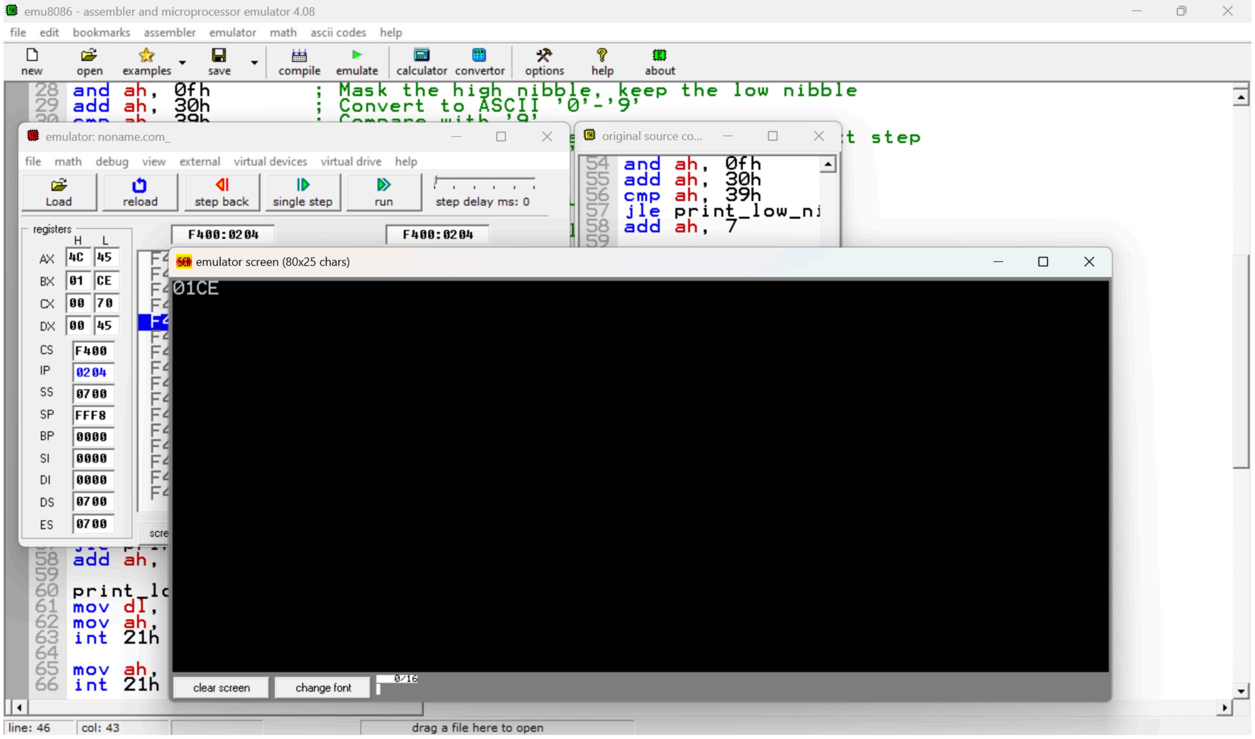
```
(DOS interrupt 21h, function 02h)
II
ack to AH
keep lower
0'-'9'
to '9', skip next step
A'-'F' (if necessary)

print_sec_digit:
mov dl, ah                 ; Move second digit to DL
mov ah, 02h                ; Prepare for output
int 21h                    ; Print second digit

; Terminate the program
mov ah, 4Ch                ; Prepare for program termination
int 21h                    ; Terminate program
```

Write a program in assembly language to perform multiplication of 16-bit data.

```
org 100h          ; Set starting address

; Load two 16-bit values into AX and BX
mov ax, 0015h      ; Load AX with first number
mov bx, 0016h      ; Load BX with second number

; Multiply AX by BX
mul bx            ; Multiply AX by BX, result in DX:AX

; Move the lower 16 bits of the result (AX) into BX
mov bx, ax         ; BX now holds the result's lower 16 bits

; --- Convert and print the high nibble of BH (BX high byte) ---
mov ah, bh         ; Move BH (high byte of BX) to AH
shr ah, 4          ; Shift right to isolate the high nibble
add ah, 30h        ; Convert to ASCII '0'-'9'
cmp ah, 39h        ; Compare with '9'
jle print_high_nibble ; If less than or equal to '9', skip next step
add ah, 7          ; Adjust to ASCII 'A'-'F' if needed

print_high_nibble:
mov dl, ah         ; Move the ASCII value to DL
mov ah, 02h        ; Prepare for output (DOS function 02h)
int 21h           ; Print the high nibble of BH

; --- Convert and print the low nibble of BH (BX high byte) ---
mov ah, bh         ; Move BH back to AH
and ah, 0fh        ; Mask the high nibble, keep the low nibble
add ah, 30h        ; Convert to ASCII '0'-'9'
cmp ah, 39h        ; Compare with '9'
jle print_low_nibble ; If less than or equal to '9', skip next step
add ah, 7          ; Adjust to ASCII 'A'-'F' if needed

print_low_nibble:
mov dl, ah         ; Move the ASCII value to DL
mov ah, 02h        ; Prepare for output
int 21h           ; Print the low nibble of BH

; --- Convert and print the high nibble of BL (BX low byte) ---
mov ah, bl         ; Move BL (low byte of BX) to AH
shr ah, 4          ; Shift right to isolate the high nibble
add ah, 30h        ; Convert to ASCII '0'-'9'
cmp ah, 39h        ; Compare with '9'
```

```
jle print_high_nibble2 ; If less than or equal to '9', skip next step
add ah, 7          ; Adjust to ASCII 'A'-'F' if needed

print_high_nibble2:
mov dl, ah         ; Move the ASCII value to DL
mov ah, 02h        ; Prepare for output
int 21h            ; Print the high nibble of BL

; --- Convert and print the low nibble of BL (BX low byte) ---
mov ah, bl         ; Move BL back to AH
and ah, 0fh        ; Mask the high nibble, keep the low nibble
add ah, 30h        ; Convert to ASCII '0'-'9'
cmp ah, 39h        ; Compare with '9'
jle print_low_nibble2 ; If less than or equal to '9', skip next step
add ah, 7          ; Adjust to ASCII 'A'-'F' if needed

print_low_nibble2:
mov dl, ah         ; Move the ASCII value to DL
mov ah, 02h        ; Prepare for output
int 21h            ; Print the low nibble of BL

mov ah, 4Ch        ; DOS terminate function
int 21h            ; Terminate the program
```

OUTPUT:



emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
emulator screen (80x25 chars)
clear screen change font
line: 46 col: 43 drag a file here to open

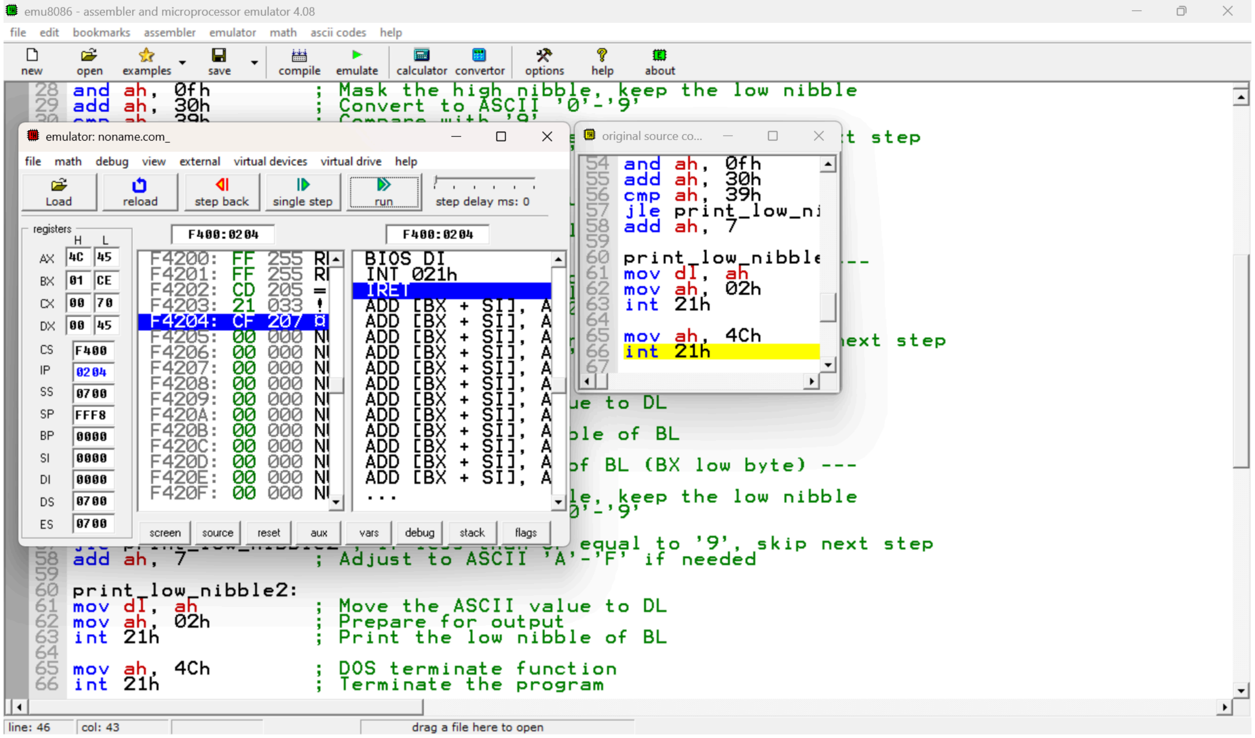
```
and ah, 0fh               ; Mask the high nibble, keep the low nibble
add ah, 30h               ; Convert to ASCII '0'-'9'
```

```
and ah, 0fh
add ah, 30h
cmp ah, 39h
jle print_low_ni
add ah, 7
```

```
F400:0204    F400:0204
```

registers	H	L
AX	4C	45
BX	01	CE
CX	00	70
DX	00	45
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

```
01CE
```



emu8086 - assembler and microprocessor emulator 4.08

```
and ah, 0fh                ; Mask the high nibble, keep the low nibble
add ah, 30h                ; Convert to ASCII '0'-'9'
```

emulator: noname.com_

step delay ms: 0

registers	H	L
AX	4C	45
BX	01	CE
CX	00	70
DX	00	45
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204

```
F4200: FF 255 RI
F4201: FF 255 RI
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ¤
F4205: 00 000 NI
F4206: 00 000 NI
F4207: 00 000 NI
F4208: 00 000 NI
F4209: 00 000 NI
F420A: 00 000 NI
F420B: 00 000 NI
F420C: 00 000 NI
F420D: 00 000 NI
F420E: 00 000 NI
F420F: 00 000 NI
```

F400:0204

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
...
```

original source co...

```
and ah, 0fh
add ah, 30h
cmp ah, 39h
jle print_low_ni
add ah, 7

print_low_nibble
mov dl, ah
mov ah, 02h
int 21h

mov ah, 4Ch
int 21h
```

```
t step
ext step
ue to DL
ble of BL
of BL (BX low byte) ---
le, keep the low nibble
0'-'9'
equal to '9', skip next step
add ah, 7                  ; Adjust to ASCII 'A'-'F' if needed

print_low_nibble2:
mov dl, ah                 ; Move the ASCII value to DL
mov ah, 02h                ; Prepare for output
int 21h                    ; Print the low nibble of BL

mov ah, 4Ch                ; DOS terminate function
int 21h                    ; Terminate the program
```

line: 46 col: 43 drag a file here to open