16 BIT HEXADECIMAL ADDITION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789ABCDEF'
n1      dw      05555h
n2      dw      05555h
result  dw      00000h
msg     db      'the result is '
asciir  db      4 dup(?)
db      cr,lf,'$'
data    ends

code    segment
assume cs:code,ds:data
start:  mov     ax,data
mov     ds,ax
mov     ax,n1
add     ax,n2
lea     bx,table
mov     result,ax
lea     si,asciir
add     si,3
mov     ax,result
and     ax,0000Fh
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,000f0h
mov     cl,04h
shr     al,cl
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,00f00h
mov     cl,08h
shr     ax,cl
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,0f000h
mov     cl,0ch
shr     ax,cl
xlat
mov     [si],al
mov     ah,09h
lea     dx,msg
```

```
int     21h


quit:   mov     al,00h
mov     ah,04ch
int     21h
code    ends
end     start
```

```
C:\BIN>16add
the result is AAAA

C:\BIN>
```

16 BIT HEXADECIMAL SUBTRACTION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789ABCDEF'
n1      dw      05555h
n2      dw      03333h
result  dw      00000h
msg     db      'the result is '
asciir  db      4 dup(?)
db      cr,lf,'$'
data    ends

code    segment
assume cs:code,ds:data
start:  mov     ax,data
mov     ds,ax
mov     ax,n1
sub     ax,n2
lea     bx,table
mov     result,ax
lea     si,asciir
add     si,3
mov     ax,result
and     ax,0000Fh
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,000f0h
mov     cl,04h
shr     al,cl
```

```
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,00f00h
mov     cl,08h
shr     ax,cl
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,0f000h
mov     cl,0ch
shr     ax,cl
xlat
mov     [si],al
mov     ah,09h
lea     dx,msg
int     21h


quit:   mov     al,00h
mov     ah,04ch
int     21h
code    ends
end     start
```

16 BIT HEXADECIMAL MULTIPLICATION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789ABCDEF'
n1      db      00005h
n2      db      00004h
result  dw      00000h
msg     db      'the result is '
asciir  db      4 dup(?)
db      cr,lf,'$'
```

```
data    ends

code    segment
assume cs:code,ds:data
start:  mov     ax,data
mov     ds,ax
mov     al,n1
mov     bl,n2
mul     bl
lea     bx,table
mov     result,ax
lea     si,asciir
add     si,3
mov     ax,result
and     ax,0000Fh
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,000f0h
mov     cl,04h
shr     al,cl
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,00f00h
mov     cl,08h
shr     ax,cl
xlat
mov     [si],al
dec     si
mov     ax,result
and     ax,0f000h
mov     cl,0bh
shr     al,cl
xlat
mov     [si],al
mov     ah,09h
lea     dx,msg
int     21h
quit:   mov     al,00h
mov     ah,04ch
int     21h
code    ends
end     start
```

```
C:\BIN>16bitmul
the result is 0014

C:\BIN>
```

32 BIT BY 16 BIT HEXADECIMAL DIVISION:

```
CR EQU 0DH
LF EQU 0AH
DATA SEGMENT
TABLE DB '0123456789ABCDEF'
LSBDIVIDEND DW 00005H
MSBDIVIDEND DW 00000H
DIVISOR DW 00002H
REMAINDER DW 00H
QUOTIENT DW 00H
MESSAGE1 DB ' THE QUOTIENT IS:'
ASCIIQUOTIENT DB 4 dup(?)
DB CR, LF, '$'
MESSAGE2 DB ' THE REMAINDER IS:'
ASCIIREMAINDER DB 4 dup(?)
DB CR, LF, '$'
DATA ENDS
CODE SEGMENT
Assume CS:Code, DS: Data
START: MOV AX, DATA
MOV DS, AX
MOV AX, LSBDIVIDEND
MOV DX, MSBDIVIDEND
DIV DIVISOR
MOV QUOTIENT, AX
MOV REMAINDER, DX
LEA BX, TABLE
LEA SI, ASCIIQUOTIENT
ADD SI,3
MOV AX, QUOTIENT
AND AX, 0000FH
XLAT
MOV [SI], AL
DEC SI
MOV AX, QUOTIENT
AND AL,000F0H
MOV CL,04H
SHR AL, CL
XLAT
MOV [SI], AL
DEC SI
MOV AX, QUOTIENT
AND AX,00F00H
MOV CL,08
SHR AX, CL
XLAT
MOV [SI], AL
DEC SI
MOV AX, QUOTIENT
AND AX,0F000H
```

```
MOV CL,0CH
SHR AX, CL
XLAT
MOV [SI], AL
MOV AH, 09H
LEA DX, MESSAGE1
INT 21H
LEA SI, ASCIIREMAINDER
ADD SI,3
MOV AX, REMAINDER
AND AX, 0000FH
XLAT
MOV [SI], AL
DEC SI
MOV AX, REMAINDER
AND AL,000F0H
MOV CL,04H
SHR AL, CL
XLAT
MOV [SI], AL
DEC SI
MOV AX, REMAINDER
AND AX,00F00H
MOV CL,08
SHR AX, CL
XLAT
MOV [SI], AL
DEC SI
MOV AX, REMAINDER
AND AX,0F000H
MOV CL,0CH
SHR AX, CL
XLAT
MOV [SI], AL
MOV AH, 09H
LEA DX, MESSAGE2
INT 21H
QUIT: MOV AL,0
MOV AH,04CH
INT 21H
CODE ENDS
END START
```

8 BIT HEXADECIMAL ADDITION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789'
n1      db      002h
n2      db      002h
res     db      00h
msg     db      'The Result is '
asres   db      2 dup(?)
        db      cr,lf,'$'
data    ends

code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     al,n1
        add     al,n2
        mov     res,al
        lea     bx,table
        lea     si,asres
        inc     si
        mov     al,res
        and     al,00FH
        xlat
        mov     [si],al
        dec     si
        mov     al,res
        and     al,0F0H
        mov     cl,04H
        shr     al,cl
        xlat
        mov     [si],al
        mov     ah,09H
        lea     dx,msg
        int     21h
quit:   mov     al,0
        mov     ah,04ch
        int     21h
        code    ends
        end     start
```

8 BIT HEXADECIMAL SUBTRACTION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789ABCDEF'
n1      db      002h
n2      db      002h
res     db      00h
msg     db      'The Result is '
asres   db      2 dup(?)
        db      cr,lf,'$'
data    ends

code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     al,n1
        sub     al,n2
        mov     res,al
        lea     bx,table
        lea     si,asres
        inc     si
        mov     al,res
        and     al,00FH
        xlat
        mov     [si],al
        dec     si
        mov     al,res
        and     al,0F0H
        mov     cl,04H
        shr     al,cl
        xlat
        mov     [si],al
        mov     ah,09H
        lea     dx,msg
        int     21h
quit:   mov     al,0
        mov     ah,04ch
        int     21h
        code    ends
        end     start
```

8 BIT HEXADECIMAL MULTIPLICATION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789ABCDEF'
n1      db      005h
n2      db      005h
result  dw      00000h
msg     db      'the result is '
asciir  db      4 dup(?)
        db      cr,lf,'$'
data    ends

code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     al,n1
        mul     n2
        lea     bx,table
        mov     result,ax
        lea     si,asciir
        add     si,3
        mov     ax,result
        and     ax,0000Fh
        xlat
        mov     [si],al
        dec     si
        mov     ax,result
        and     ax,000f0h
        mov     cl,04h
        shr     al,cl
        xlat
        mov     [si],al
        dec     si
        mov     ax,result
        and     ax,00f00h
        mov     cl,08h
        shr     ax,cl
        xlat
        mov     [si],al
        dec     si
        mov     ax,result
        and     ax,0f000h
        mov     cl,0ch
        shr     ax,cl
        xlat
        mov     [si],al
        mov     ah,09h
        lea     dx,msg
```

```
        int     21h


quit:   mov     al,00h
        mov     ah,04ch
        int     21h
code    ends
        end     start
```

16 BIT BY 8 BIT HEXADECIMAL DIVISION:

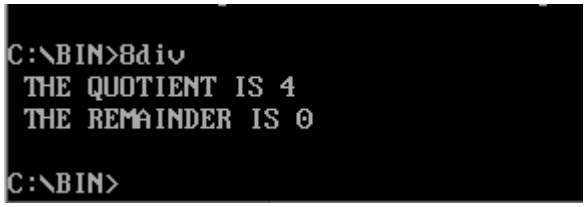
```
CR              EQU     0DH
LF              EQU     0AH

DATA            SEGMENT
TABLE           DB      '0123456789'
LSBDIVIDENT     DB      08
MSBDIVIDENT     DB      00
DIVISOR                 DB      02
MESSAGE1        DB      ' THE QUOTIENT IS '
ASCIIQUOTIENT           DB      1 dup(?)
                DB      CR, LF, '$'
MESSAGE2        DB      ' THE REMAINDER IS '
ASCIIREMAINDER          DB      1 dup(?)
                DB      CR, LF, '$'

DATA            ENDS

CODE            SEGMENT
                Assume          CS:Code, DS: Data
START:          MOV     AX, DATA
                MOV     DS, AX
                MOV     AL, LSBDIVIDENT
                MOV     AH, MSBDIVIDENT
                MOV     DH,DIVISOR
                DIV     DH
                LEA     BX,TABLE
                LEA     SI, ASCIIQUOTIENT
                XLAT
                MOV     [SI], AL
                MOV     AH,09H
                LEA     DX,MESSAGE1
```

```
        INT     21H
        LEA     SI, ASCIIREMAINDER
        MOV             AL,DH
        XLAT
        MOV     [SI], AL
        MOV     AH,09H
        LEA     DX, MESSAGE2
        INT     21H
QUIT:   MOV     AL,0
        MOV     AH,04CH
        INT     21H
        CODE    ENDS
        END     START
```



8 BIT DECIMAL ADDITION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789'
n1      db      099h
n2      db      099h
res     db      00h
msg     db      'The Result is '
asres   db      2 dup(?)
        db      cr,lf,'$'
data    ends

code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     al,n1
        add     al,n2
        daa
        mov     res,al
        lea     bx,table
        lea     si,asres
        inc     si
        mov     al,res
        and     al,00FH
        xlat
```

```
        mov     [si],al
        dec     si
        mov     al,res
        and     al,0F0H
        mov     cl,04H
        shr     al,cl
        xlat
        mov     [si],al
        mov     ah,09H
        lea     dx,msg
        int     21h
quit:   mov     al,0
        mov     ah,04ch
        int     21h
        code    ends
        end     start
```

```
C:\BIN>8decadd
The Result is 98

C:\BIN>_
```

8 BIT DECIMAL SUBTRACTION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789ABCDEF'
n1      db      099h
n2      db      099h
res     db      00h
msg     db      'The Result is '
asres   db      2 dup(?)
        db      cr,lf,'$'
data    ends

code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     al,n1
        sub     al,n2
        das
        mov     res,al
        lea     bx,table
        lea     si,asres
        inc     si
        mov     al,res
        and     al,00FH
```

```
        xlat
        mov     [si],al
        dec     si
        mov     al,res
        and     al,0F0H
        mov     cl,04H
        shr     al,cl
        xlat
        mov     [si],al
        mov     ah,09H
        lea     dx,msg
        int     21h
quit:   mov     al,0
        mov     ah,04ch
        int     21h
        code    ends
        end     start
```

8 BIT DECIMAL MULTIPLICATION OF TWO NUMBERS:

```
cr      equ     0dh
lf      equ     0ah

data    segment
table   db      '0123456789'
n1      db      04h
n2      db      04h
msg     db      ' The Result is '
asres   db      2 dup(?)
        db      cr,lf,'$'
data    ends

code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     al,n1
        mul     n2
        aam
        lea     bx,table
        lea     si,asres
        inc     si
        and     al,00FH
```

```
        xlat
        mov     [si],al
        dec     si
        mov     al,ah
        and     al,00FH
        xlat
        mov     [si],al
        mov     ah,09H
        lea     dx,msg
        int     21h
quit:   mov     al,0
        mov     ah,04ch
        int     21h
        code    ends
        end     start
```

16 BY 8 BIT DECIMAL DIVISION:

```
CR EQU 0DH
LF EQU 0AH

DATA SEGMENT
TABLE   DB '0123456789ABCDF'
lsbdiv  db      04h
msbdiv  db      00h
divisor db      02h
msg1    db      ' The Quotient is '
asquo   db      2 dup(?)
db      cr,lf,'$'
msg2    db      ' The Remainder is '
asrem   db      2 dup(?)
db      cr,lf,'$'

data    ends

code    segment
assume cs:code,ds:data
start:  mov     ax,data
mov     ds,ax
mov     al,lsbdiv
```

```
mov     ah,msbdiv
mov     dh,divisor
aad
div     dh
lea     bx,table
lea     si,asquo
xlat
mov     [si],al
mov     ah,09h
lea     dx,msg1
int     21h
lea     bx,table
lea     si,asrem
mov al,dh
xlat
mov     [si],al
mov     ah,09h
lea     dx,msg2
int     21h


quit:   mov     al,0
mov     ah,04ch
int     21h
code ends
end     start
```