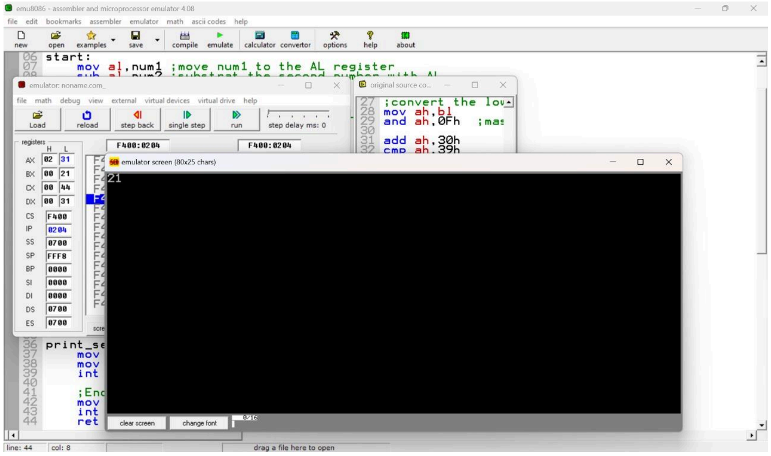
Program in assembly language to perform subtraction of 8-bit data.

```
org 100h

num1 db 39h
num2 db 18h

start:
   mov al,num1 ;move num1 to the AL register
   sub al,num2 ;substrat the second number with AL

   mov bl,al

;convert the upper nibble(4 bits ) of AL to characters
   mov ah,al
   and ah,0F0h ;mask the lower nibble
   shr ah,4

   add ah,30h ;convert the ASCHII digit(0-9)
   cmp ah,39h
   jle print_first_digit
   add ah,7   ;convert to ASCHII letter(A-F) if necessary

print_first_digit:
   mov dl,ah ;move the first digit to DL for printing
   mov ah,02h  ;BIOS interrupt to display charater
   int 21h

;convert the lower nibble(4 bits) of AL to characters
   mov ah,bl
   and ah,0Fh  ;mask the upper nibble

   add ah,30h
   cmp ah,39h
   jle print_second_digit
   add ah,7

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

   ;End the program
   mov ah,4Ch
```

```
    int 21h
    ret
```

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: 44 col: 8 drag a file here to open

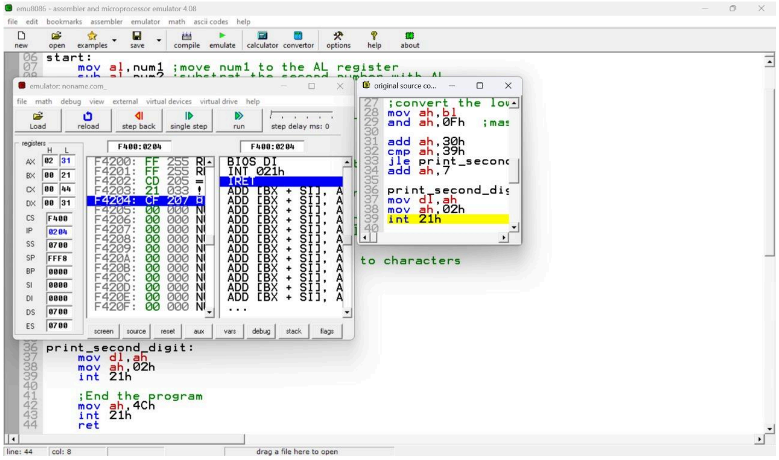
```
start:
    mov al,num1 ;move num1 to the AL register
```

```
;convert the low
mov ah,bl
and ah,0Fh   ;mas
add ah,30h
cmp ah,39h
```

```
F400:0204    F400:0204
registers
     H  L
AX  02 31
BX  00 21
CX  00 44
DX  00 31
CS  F400
IP  0204
SS  0700
SP  FFF8
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
21
```

```
print_se
    mov
    mov
    int

    ;End
    mov
    int
    ret
```


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...
screen source reset aux vars debug stack flags
line: 44 col: 8 drag a file here to open

```
start:
    mov al,num1 ;move num1 to the AL register
```

```
F400:0204    F400:0204
registers
     H  L
AX  02 31
BX  00 21
CX  00 44
DX  00 31
CS  F400
IP  0204
SS  0700
SP  FFF8
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
F4200:  FF 255 RI     BIOS DI
F4201:  FF 255 RI     INT 021h
F4202:  CD 205 =      IRET
F4203:  21 033 !      ADD [BX + SI], A
F4204:  CF 207 ¤      ADD [BX + SI], A
F4205:  00 000 NI     ADD [BX + SI], A
F4206:  00 000 NI     ADD [BX + SI], A
F4207:  00 000 NI     ADD [BX + SI], A
F4208:  00 000 NI     ADD [BX + SI], A
F4209:  00 000 NI     ADD [BX + SI], A
F420A:  00 000 NI     ADD [BX + SI], A
F420B:  00 000 NI     ADD [BX + SI], A
F420C:  00 000 NI     ADD [BX + SI], A
F420D:  00 000 NI     ADD [BX + SI], A
F420E:  00 000 NI     ADD [BX + SI], A
F420F:  00 000 NI     ...
```

```
;convert the low
mov ah,bl
and ah,0Fh   ;mas

add ah,30h
cmp ah,39h
jle print_seconc
add ah,7

print_second_dig
mov dl,ah
mov ah,02h
int 21h
```

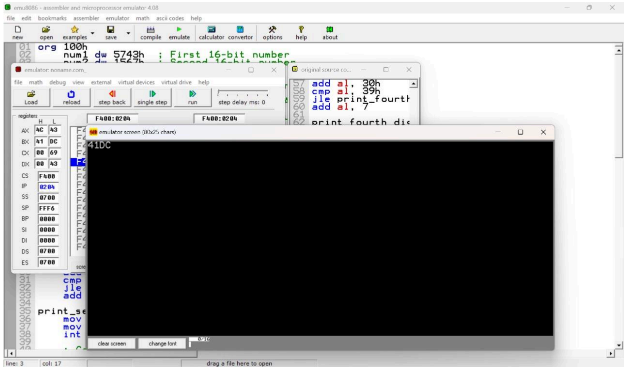
```
to characters
```

```
print_second_digit:
    mov dl,ah
    mov ah,02h
    int 21h

    ;End the program
    mov ah,4Ch
    int 21h
    ret
```

Program in assembly language to perform subtraction of 16-bit data.

```
org 100h
    num1 dw 5743h  ; First 16-bit number
    num2 dw 1567h  ; Second 16-bit number

start:
    ; Load the lower bytes of num1 and num2
    mov ax, num1     ; Load num1 into AX (AX = 1234h)
    sub ax, num2     ; Add num2 to AX (AX = AX + num2)

    ; Store the result in BX for later use
    mov bx, ax         ; Copy AX to BX

    ; Convert upper byte (high 8 bits) to ASCII and display
    mov al, ah          ; Move the high byte of AX to AL
    and al, 0F0h        ; Mask the lower nibble
    shr al, 4           ; Shift right to get the upper nibble
    add al, 30h         ; Convert to ASCII digit
    cmp al, 39h         ; Compare with ASCII value of '9'
    jle print_first_digit
    add al, 7           ; Convert to ASCII letter if needed

print_first_digit:
    mov dl, al          ; Move AL to DL for printing
    mov ah, 02h         ; BIOS interrupt to display character
    int 21h

    ; Convert lower nibble of the high byte to ASCII and display
    mov al, bh          ; Move the high byte of BX to AL again
    and al, 0Fh         ; Mask the upper nibble
    add al, 30h         ; Convert to ASCII digit
    cmp al, 39h         ; Compare with ASCII value of '9'
    jle print_second_digit
    add al, 7           ; Convert to ASCII letter if needed

print_second_digit:
    mov dl, al          ; Move AL to DL for printing
    mov ah, 02h         ; BIOS interrupt to display character
    int 21h

    ; Convert upper nibble of the low byte to ASCII and display
    mov al, bl          ; Move the low byte of BX to AL
    and al, 0F0h        ; Mask the lower nibble
    shr al, 4           ; Shift right to get the upper nibble
    add al, 30h         ; Convert to ASCII digit
    cmp al, 39h         ; Compare with ASCII value of '9'
    jle print_third_digit
    add al, 7           ; Convert to ASCII letter if needed

print_third_digit:
    mov dl, al          ; Move AL to DL for printing
    mov ah, 02h         ; BIOS interrupt to display character
    int 21h

    ; Convert lower nibble of the low byte to ASCII and display
    mov al, bl          ; Move the low byte of BX to AL
    and al, 0Fh         ; Mask the upper nibble
    add al, 30h         ; Convert to ASCII digit
    cmp al, 39h        ; Compare with ASCII value of '9'
    jle print_fourth_digit
    add al, 7          ; Convert to ASCII letter if needed

print_fourth_digit:
    mov dl, al         ; Move AL to DL for printing
    mov ah, 02h        ; BIOS interrupt to display character
    int 21h

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

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: 3 col: 17 drag a file here to open

```
org 100h
    num1 dw 5743h   ; First 16-bit number
```

```
add al, 30h
cmp al, 39h
jle print_fourth
add al, 7

print_fourth_di
```

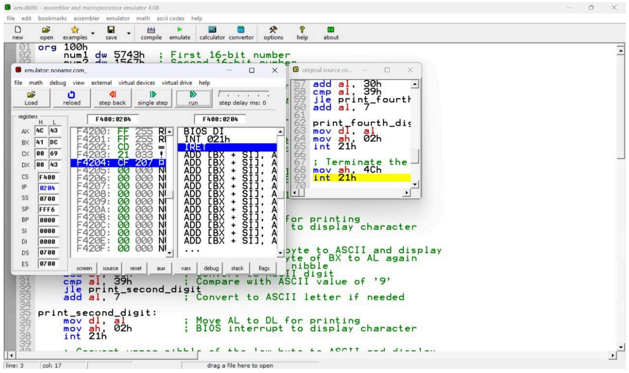
```
41DC
```

registers

	H	L
AX	4C	43
BX	41	DC
CX	00	69
DX	00	43
CS	F400	
IP	0204	
SS	0700	
SP	FFF6	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204 F400:0204

```
    cmp
    jle
    add

print_se
    mov
    mov
    int
```



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
screen source reset aux vars debug stack flags
original source co...
line: 3 col: 17 drag a file here to open

```
org 100h
    num1 dw 5743h   ; First 16-bit number
```

registers

	H	L
AX	4C	43
BX	41	DC
CX	00	69
DX	00	43
CS	F400	
IP	0204	
SS	0700	
SP	FFF6	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204 F400:0204

```
F4200: FF 255 RI     BIOS DI
F4201: FF 255 RI     INT 021h
F4202: CD 205 =      IRET
F4203: 21 033 !      ADD [BX + SI], A
F4204: CF 207 ¤      ADD [BX + SI], A
F4205: 00 000 NI     ADD [BX + SI], A
F4206: 00 000 NI     ADD [BX + SI], A
F4207: 00 000 NI     ADD [BX + SI], A
F4208: 00 000 NI     ADD [BX + SI], A
F4209: 00 000 NI     ADD [BX + SI], A
F420A: 00 000 NI     ADD [BX + SI], A
F420B: 00 000 NI     ADD [BX + SI], A
F420C: 00 000 NI     ADD [BX + SI], A
F420D: 00 000 NI     ADD [BX + SI], A
F420E: 00 000 NI     ADD [BX + SI], A
F420F: 00 000 NI     ...
```

```
add al, 30h
cmp al, 39h
jle print_fourth
add al, 7

print_fourth_di
mov dl, al
mov ah, 02h
int 21h

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

```
                 for printing
                  to display character

                 byte to ASCII and display
                 yte of BX to AL again
                  nibble
                 II digit
    cmp al, 39h          ; Compare with ASCII value of '9'
    jle print_second_digit
    add al, 7            ; Convert to ASCII letter if needed

print_second_digit:
    mov dl, al           ; Move AL to DL for printing
    mov ah, 02h          ; BIOS interrupt to display character
    int 21h
```