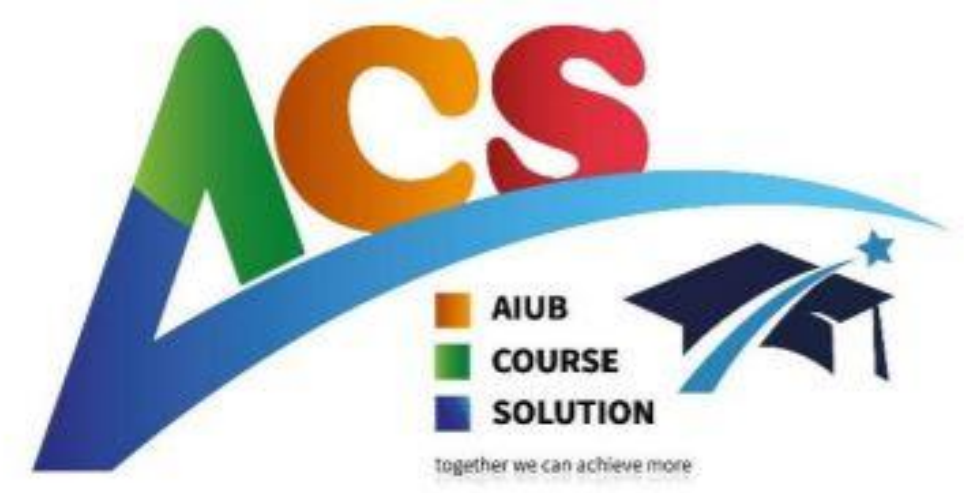

COMPUTER ORGANIZATION AND ARCHITECTURE

Solved by:

SIFAT SHAHARIA

Chapter 4

Ex:1

```
.MODEL SMALL
.STACK 100H
.DATA
MSG DB 'THE SUM OF '
V1 DB ?
M1 DB ' AND '
V2 DB ?
M2 DB ' IS '
R1 DB ?
M3 DB '$'

.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX

MOV AH,2
MOV BL,'?'     ;PRINT ? MARK
MOV DL,BL
INT 21H
```

```
MOV AH,1
INT 21H       ;INPUT 1ST DIGIT
MOV V1,AL

INT 21H       ;INPUT 2ND DIGIT
MOV V2,AL

MOV CL,V1
ADD CL,V2     ;ADD OPERATION
SUB CL,48

MOV R1,CL     ;PASS RESULT IN MSG
MOV AH,2
MOV DL,10     ;NEW LINE
INT 21H
MOV DL,13
INT 21H
LEA DX,MSG    ;PRINT MSG
MOV AH,9
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

EX:2

```
.MODEL SMALL
.STACK
.DATA
MSG1 DB 'ENTER THREE INITIALS: $'

.CODE

MAIN PROC

MOV AX,@DATA
MOV DS,AX

LEA AX,MSG1
MOV AH,9      ;PRINT MSG1
INT 21H

MOV AH,1
INT 21H       ;INPUT 1ST DIGIT
MOV BL,AL

INT 21H
MOV CL,AL     ;INPUT 2ND DIGIT

INT 21H
```

```
MOV BH,AL     ;INPUT 3RD DIGIT


MOV AH,2
MOV DL,10
INT 21H       ;NEW LINE
MOV DL,13
INT 21H

MOV DL,BL     ;PRINT 1ST DIGIT
INT 21H

MOV AH,2
MOV DL,10     ;NEW LINE
INT 21H
MOV DL,13
INT 21H

MOV DL,CL     ;PRINT 2ND DIGIT
INT 21H

MOV AH,2
MOV DL,10     ;NEW LINE
INT 21H
MOV DL,13
INT 21H
```

```
MOV DL,BH    ;PRINT 3RD DIGIT
INT 21H


MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

EX: 3

```
.MODEL SMALL
.STACK 100H
.DATA
MSG DB 'ENTER A HEX VALUE DIGIT: '

V1 DB ?
M3 DB 10,13,'THE DECIMAL VALUE OF '
R1 DB ?
M4 DB ' IS: 1$'
.CODE
MAIN PROC
```

```
MOV AX,@DATA
MOV DS,AX

MOV AH,1
INT 21H        ;INPUT THE HEX VALUE
MOV V1,AL

MOV AH,2
MOV DL,10
INT 21H        ;NEW LINE
MOV DL,13
INT 21H

MOV CL,V1      ;PASS THE CHAR INTO R1
MOV R1,CL

MOV BL,R1
SUB BL,17      ;CONVERTED INTO DECIMAL

LEA DX,MSG
MOV AH,9       ;PRINT MSG
INT 21H

MOV AH,2
MOV DL,BL
```

```
INT 21H



MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

EX:4

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB '##########',10,13,'$'
.CODE
MAIN PROC

MOV AX,@DATA
MOV DS,AX

LEA DX,MSG1      ;PRINT MSG1
MOV AH,9
```

INT 21H

LEA DX,MSG1 ;PRINT MSG1

INT 21H

LEA DX,MSG1 ;PRINT MSG1

INT 21H

LEA DX,MSG1

;PRINT MSG1

INT 21H

LEA DX,MSG1

;PRINT MSG1

INT 21H

LEA DX,MSG1

;PRINT MSG1

```
INT 21H



LEA DX,MSG1
;PRINT MSG1
INT 21H



LEA DX,MSG1
;PRINT MSG1
INT 21H



LEA DX,MSG1     ;PRINT MSG1

INT 21H



LEA DX,MSG1
;PRINT MSG1
INT 21H

MOV AH,4CH
INT 21H
MAIN ENDP
```

```
END MAIN
```

EX:5

```
.MODEL SMALL
.STACK 100H
.DATA

MSG1 DB '#######',10,13,'$'
MSG2 DB '##$'
MSG3 DB '##',10,13,'$'
MSG4 DB 'ENTER THREE INTITIALS: $'

.CODE

MAIN PROC
MOV AX,@DATA
MOV DS,AX

LEA DX,MSG4
MOV AH,9        ;PRINT MSG4
INT 21H

MOV AH,1
INT 21H         ;INPUT 1ST DIGIT
MOV BL,AL
```

```
INT 21H         ;INPUT 2ND DIGIT
MOV CL,AL

INT 21H         ;INPUT 3RD DIGIT
MOV BH,AL

MOV AH,2
MOV DL,10
INT 21H         ;NEW LINE
MOV DL,13
INT 21H

LEA DX,MSG1     ;PRINT MSG1
MOV AH,9
INT 21H



LEA DX,MSG1     ;PRINT MSG1
INT 21H

LEA DX,MSG1     ;PRINT MSG1
INT 21H
```

```
LEA DX,MSG2      ;PRINT MSG2
INT 21H

MOV AH,2
MOV DL,BL        ;PRINT 1ST DIGIT
INT 21H

MOV DL,CL
INT 21H          ;PRINT 2ND DIGIT

MOV DL,BH
INT 21H          ;PRINT 3RD DIGIT

LEA DX,MSG3
MOV AH,9         ;PRINT MSG3
INT 21H

LEA DX,MSG1      ;PRINT MSG1
INT 21H

LEA DX,MSG1      ;PRINT MSG1
INT 21H

LEA DX,MSG1      ;PRINT MSG1
INT 21H
```

```
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

ENTER 2 DIGIT AND SUM OF THEM

```
.model small
.stack 100h
.data
msg1 db 'enter first digit:  $'
msg2 db 'enter second digit:  $'
msg3 db 'the sum is = $'
.code

main proc
    mov ax,@data
    mov ds,ax


    lea dx,msg1      ;print msg1
    mov ah,9
    int 21h
```

```
mov ah,1
int 21h         ;input 1st digit
mov bl,al

mov ah,2
mov dl,0ah
int 21h         ;new line
mov dl,0dh
int 21h

lea dx,msg2     ;print msg2
mov ah,9
int 21h

mov ah,1
int 21h         ;input 2nd digit
mov cl,al

mov ah,2
mov dl,0ah      ;new line
int 21h
mov dl,0dh
int 21h
```

```
    lea dx,msg3      ;print msg3
    mov ah,9
    int 21h



    add bl,cl       ;add operation
    sub bl,58



    mov ah,2        ;print 1 by default
    mov dl,'1'
    int 21h
    mov dl,bl       ;show result
    int 21h



    mov ah,4ch
    int 21h
    main endp
end main
```

```
.model small
.stack 100h
.data
msg1 db 'enter first digit:  $'
msg2 db 'enter second digit:  $'
msg3 db 'the sum is = $'
.code

main proc
   mov ax,@data
   mov ds,ax



   lea dx,msg1      ;print msg1
   mov ah,9
   int 21h

   mov ah,1
   int 21h          ;input 1st digit
   mov bl,al

   mov ah,2
   mov dl,0ah
   int 21h          ;new line
   mov dl,0dh
```

```
int 21h


lea dx,msg2      ;print msg2
mov ah,9
int 21h


mov ah,1
int 21h          ;input 2nd digit
mov cl,al


mov ah,2
mov dl,0ah       ;new line
int 21h
mov dl,0dh
int 21h


lea dx,msg3      ;print msg3
mov ah,9
int 21h



add bl,cl       ;add operation
sub bl,58
```

```
    mov ah,2        ;print 1 by default
    mov dl,'1'
    int 21h
    mov dl,bl       ;show result
    int 21h



    mov ah,4ch
    int 21h
    main endp
end main
```

Chapter 6

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
mov ah,2
mov dl,'?'
int 21h

mov ah,1
mov bl,al
```

```
int 21h


mov ah,1
mov cl,al
int 21h


cmp bl,cl
jl small
jmp bigge
small:
mov ah,2
mov dl,bl
int 21h
bigger:
mov ah,2
mov dl,cl
int 21h
end main
```

EX: 10

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'ENTER A HEX DIGIT: $'
MSG2 DB 'IN DECIMAL IS IT: $'
MSG3 DB 'DO YOU WANT TO DO IT AGAIN? $'
MSG4 DB 'IILEGAL CHARACTER: ENTER 0-9 OR A-F: $'
MSG5 DB 10,13,'$'
.CODE

MAIN PROC
MOV AX,@DATA
MOV DS,AX
```

```
START1:
MOV AH,9
LEA DX,MSG1  ;PRINT MSG1
INT 21H

START2:
MOV AH,1
INT 21H      ;INPUT
MOV BL,AL

MOV AH,9
LEA DX,MSG5  ;NEWLINE
INT 21H


CMP BL,"0"
JB NO        ;SET NUMBER 0-9
CMP BL,"9"
JA PROGRAM    ;JUMP IF ABOVE

JMP NUMBER


PROGRAM:
CMP BL,"A"
```

```
JB NO        ;JUMP IF BELOW
CMP BL,"F"
JA NO
JMP LETTER    ;JUMP LETTER


NO:
MOV AH,9
LEA DX,MSG4   ;PRINT MSG4
INT 21H
JMP START2    ;JUMP START2


NUMBER:      ;LOOP NUMBER
MOV AH,9
LEA DX,MSG2   ;PRINT MSG2
INT 21H

MOV AH,9
LEA DX,MSG5   ;PRINT MSG5
INT 21H

JMP REPEAT    ;JUMP REPEAT
```

```
REPEAT:      ; LOOP REPEAT

MOV AH,9
LEA DX,MSG3   ;PRINT MSG3
INT 21H

MOV AH,1
INT 21H
MOV BH,AL

MOV AH,9
LEA DX,MSG5   ;NEWLINE
INT 21H

CMP BH,"Y"
JE START1     ;IF PUT Y OR y THE PROGRAM WILL REPEATS
CMP BH,"y"
JE START1
JMP EXIT      ;JUMP EXIT

LETTER:      ;LOOP LETTER
MOV AH,9
LEA DX,MSG2   ;JUMP MSG2
INT 21H
```

```
MOV AH,2
MOV DL,'1'
INT 21H

SUB BL,17
MOV DL,BL
INT 21H


MOV AH,9
LEA DX,MSG5    ;NEWLINE
INT 21H
JMP REPEAT     ;JUMP REPEAT


EXIT:        ;LOOP EXIT
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

EX: 09

```
.model small
.stack 100h
.data
msg db 10,13,'$'
.code
main proc
```

```
mov ax,@data
mov ds,ax

mov bl,80h
mov bh,0
mov cx,127

display:

cmp bh,10
je newline    ;jump if equal

mov ah,2
mov dl,bl
int 21h
inc bl       ;increment bl
inc bh       ;increment bh




loop display
jmp exit
;jump ex
```

```
newline:
mov ah,9
lea dx,msg       ;print msg
int 21h          ;loop newline
mov bh,0
jmp display


exit:
mov ah,4ch       ;loop exit
int 21h
main endp
end main
```

EX: 11

```
.model small
.stack 100h
.data
msg1 db 'Enter a hex digit: $'
msg2 db 'In decimal it is: $'
msg3 db 'Do you want to do it again? $'
msg4 db 'Illegal character - enter 0-9 or A-F: $'
msg5 db 10,13,'$'
msg6 db 10,13,'Terminated ','$'
.code
main proc
mov ax,@data
mov ds,ax

mov cl,0
```

```
start1:
mov ah,9
lea dx,msg1 ;print msg1
int 21h
start2:
mov ah,1
int 21h
mov bl,al
inc cl      ;increment cl

cmp cl,3
je doomed
mov ah,9
lea dx,msg5   ;print msg5
int 21h

cmp bl,"0"
jb no
cmp bl,"9"
ja PRO        ;jump if above
jmp number     ;jump

pro: cmp bl,"A"
jb no
cmp bl,"F"
```

```
ja no         ;jump if above
jmp letter    ;jump

no:
mov ah,9
lea dx,msg4   ;print msg4
int 21h
jmp start2    ;jump

number:
mov ah,9
lea dx,msg2   ;print msg2
int 21h

mov ah,2
mov dl,bl
int 21h

mov ah,9
lea dx,msg5    ;print msg5
int 21h
jmp repeat     ;jump

repeat:
mov ah,9
```

```
lea dx,msg3     ;print msg3
int 21h


mov ah,1
int 21h
mov bh,al


mov ah,9       ;print msg5
lea dx,msg5
int 21h


cmp bh,"Y"


je start1      ;jump if equal
jmp exit       ;jump
letter:
mov ah,9
lea dx,msg2    ;print msg2
int 21h


mov ah,2
mov dl,'1'
int 21h
sub bl,17
mov dl,bl
```

```
int 21h


mov ah,9
lea dx,msg5    ;print msg5
int 21h
jmp repeat

doomed:
mov ah,9
lea dx,msg6     ;print msg6
int 21h
jmp exit

exit:
mov ah,4ch
int 21h        ;exit
main endp
end main
```

DESENDING order

```
.model small
.stack 100h
.data
m1 db 'Enter three characters :$'
```

```
m2 db 10,13,'Descending order :','$'
.code
main proc
mov ax,@data
mov ds,ax

mov ah,9
lea dx,m1    ;print msg1
int 21h

mov ah,1
int 21h
mov bl,al
int 21h      ;take three input
mov bh,al
int 21h
mov cl,al

cmp bl,'A'
jae pro      ;jump if above or equal
cmp bl,'Z'
jbe pro      ;jump if below or equal
jmp exit     ;jump exit

cmp bh,'A'
```

```
jae pro      ;jump if above or equal
cmp bh,'Z'
jbe pro      ;jump if below or equal
jmp exit     ;jump exit

cmp cl,'A'
jae pro      ;jump if above or equal
cmp cl,'Z'
jbe pro      ;jump if below or equal
jmp exit     ;jump exit
pro:
cmp bl,bh
jnge pro2    ;jump if not greater than or equal
cmp bl,cl
jnle pro2    ;jump if not less than or equal


mov ah,9
lea dx,m2    ;print m2
int 21h
mov ah,2
mov dl,cl    ;print cl
int 21h
mov dl,bl    ;print bl
int 21h
```

```
mov dl,bh     ;print bh
int 21h


pro2:
cmp bl,bh
jnle pro3      ;jump if not less than or equal
cmp bl,cl
jnge pro3      ;jump if not greater than or equal

mov ah,9
lea dx,m2      ;print m2
int 21h
mov ah,2
mov dl,bh      ;print bh
int 21h
mov dl,bl      ;print bl
int 21h
mov dl,cl      ;print cl
int 21h
pro3:
cmp bh,bl
jnge pro4      ;jump if not greater than or equal
cmp bh,cl
jnle pro4      ;jump if not less than or equal
```

```
mov ah,9
lea dx,m2       ;print m2
int 21h
mov ah,2
mov dl,cl       ;print cl
int 21h
mov dl,bh       ;print bh
int 21h
mov dl,bl       ;print bl
int 21h
pro4:
cmp bh,bl
jnle pro5       ;jump if not less than or equal
cmp bh,cl
jnge pro5       ;jump if not greater than or equal

mov ah,9
lea dx,m2       ;print m2
int 21h
mov ah,2
mov dl,bl       ;print bl
int 21h
mov dl,bh       ;print bh
int 21h
mov dl,cl       ;print cl
```

```
int 21h
pro5:
cmp cl,bh
jnge pro6       ;jump if not greater than or equal
cmp cl,bl
jnle pro6      ;jump if not less than or equal

mov ah,9
lea dx,m2      ;print m2
int 21h
mov ah,2
mov dl,bl      ;print bl
int 21h
mov dl,cl      ;print cl
int 21h
mov dl,bh      ;print bh
int 21h
pro6:
cmp cl,bh
jnle exit      ;jump if not less than or equal
cmp cl,bl
jnge exit      ;jump if not greater than or equal

mov ah,9
lea dx,m2      ;print m2
```

```
int 21h
mov ah,2
mov dl,bh      ;print bh
int 21h
mov dl,cl      ;print cl
int 21h
mov dl,bl      ;print bl
int 21h

exit:
mov ah,4ch          ;loop exit
int 21h
main endp
end main
```

Chapter 7:

Write a program that prompts the user to enter a character ,and on subsequence line prints its ASCII code in binary and the number of one bit its ACCII code:

```
.model small
.stack 100h
.data

msg1 db 10,13,'Type a character:$'
msg2 db 10,13,'The Ascii code of $'
msg3 db ' in hex is:$'
.code


main proc


mov ax,@data
mov ds,ax


input:


lea dx,msg1
```

```
mov ah,9
int 21h


mov ah,1
int 21h

mov bl,al

cmp bl,0dh
je end

lea dx,msg2
mov ah,9
int 21h

mov dl,bl
mov ah,2
int 21h

lea dx,msg3
mov ah,9
int 21h

mov cx,4
```

```
convert:

mov dl,bh
shr dl,1 ;shift left 4 times
shr dl,1
shr dl,1
shr dl,1

cmp dl,9
jbe num

add dl,55d

jmp display


num:

add dl,30h


display:
```

```
mov ah,2
int 21h

rcl bx,1 ;rotate carry left 4 times
rcl bx,1
rcl bx,1
rcl bx,1

loop convert

jmp input


end:


MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

EX: 11

```
.MODEL SMALL
.STACK 100H
```

```
.DATA
PROMPT_1  DB  'Enter the hexadecimal number ( max 4-digit ) : $'
PROMPT_2  DB  0DH,0AH,'The equivalent 16-bit binary number is : $'
ILLEGAL   DB  0DH,0AH,'Illegal hex number. Try again : $'

COUNT     DB  ?

.CODE
MAIN PROC
MOV AX, @DATA              ; initialize DS
MOV DS, AX

LEA DX, PROMPT_1           ; load and display the string PROMPT_1
MOV AH,9
INT 21H

JMP @START                  ; jump to label @START_2

@START_1:                  ; jump label
LEA DX, ILLEGAL            ; load and display the string ILLEGAL
MOV AH, 9
INT 21H

@START:                    ;
```

```
XOR BX, BX              ; clear BX
MOV COUNT, 30H          ; initialize loop counter

@START_2:               ; jump label
MOV AH, 1               ; set input function
INT 21H                 ; read a character

CMP AL, 0DH             ; compare Al with CR
JNE @SKIP               ; jump to label @SKIP if AL!=CR

CMP COUNT, 30H          ; compare COUNT with 0
JBE @START_1            ; jump to label @START_1 if COUNT<=0
JMP @END                ; jump to label @END

@SKIP:                  ; jump label

CMP AL, "A"             ; compare AL with "A"
JB @DECIMAL             ; jump to label @DECIMAL if AL<A

CMP AL, "F"             ; compare AL with "F"
JA @START_1             ; jump to label @START_1 if AL>F
ADD AL, 09H             ; add 9 to AL
JMP @OK                 ; jump to label @OK

@DECIMAL:               ; jump label
```

```
CMP AL, 30H          ; compare AL with 0
JB @START_1          ; jump to label @START_1 if AL<0

CMP AL, 39H          ; compare AL with 9
JA @START_1          ; jump to label @START_1 if AL>9

@OK:                 ; jump label

INC COUNT            ; increment the COUNT variable

AND AL, 0FH          ; convert the ascii into binary code

MOV CL, 4            ; set CL=4
SHL AL, CL           ; shift AL towards left by 4 positions

MOV CX, 4            ; set CX=4

@LOOP_1:             ; loop label
SHL AL, 1            ; shift AL towards left by 1 position
RCL BX, 1            ; rotate BX towards left by 1 position
; through carry
LOOP @LOOP_1         ; jump to label @LOOP_1 if CX!=0

CMP COUNT, 34H       ; compare COUNT with 4
JE @END              ; jump to label @END if COUNT=4
```

```
JMP @START_2            ; jump to label @START_2

@END:                   ; jump label

LEA DX, PROMPT_2        ; load and display the string PROMPT_2
MOV AH, 9
INT 21H

MOV CX, 16              ; set CX=16
MOV AH, 2               ; set output function

@LOOP_2:                ; loop label
SHL BX, 1               ; shift BX towards left by 1 position
JC @ONE                 ; jump to label @ONE if CF=1
MOV DL, 30H             ; set DL=0
JMP @DISPLAY            ; jump to label @DISPLAY

@ONE:                   ; jump label
MOV DL, 31H             ; set DL=1

@DISPLAY:               ; jump label
INT 21H                 ; display a character
LOOP @LOOP_2            ; jump to label @LOOP_2 if CX!=0

MOV AH, 4CH             ; return control to DOS
```

```
INT 21H

MAIN ENDP

END MAIN
```

Ex: 10

```
.model small

.stack 100h

.data

msg1 db 10,13,' Type a hex number (0 to FFFF):$'

msg2 db 10,13,'Illegal hex digit,try again:$'

msg3 db 10,13,'In Binary it is $'


.code


main proc


mov ax,@data

mov ds,ax




jmp input



invalid:
```

```
lea dx, msg2
mov ah,9
int 21h


input:

lea dx,msg1
mov ah,9
int 21h

xor bx,bx
mov cl,4

mov ah,1
int 21h


convert:

cmp al,0dh
je end_input
```

```
cmp al,'0'
jb invalid

cmp al,'F'
ja invalid



cmp al,39h
ja letter

and al,0fh
jmp left



letter:

sub al,55d ;convert char to binary

left:
shl bx,cl
or bl,al

mov ah,1
int 21h
jmp convert
```

```
end_input:

lea dx,msg3
mov ah,9
int 21h


xor dx,dx

mov cx,16


print_binary:
shl bx,1 ;catch bx bit

jc one ;cf=1

mov dl,30h
jmp display

one: mov dl,31h

display:
```

```
mov ah,2
int 21h
loop print_binary


mov ah,4ch
int 21h
main endp
end main
```

Ex: 11

```
.model small
.stack 100h
.data
msg1 db 'Type a binary number upto 16 digits:$'
msg2 db 10,13,'in hex it is:$'
.code


main proc

mov ax,@data
mov ds,ax


lea dx,msg1
```

```
mov ah,9
int 21h

xor bx,bx
mov ah,1
int 21h

input:

cmp al,0dh
je exit



and al,0fh
shl bx,1
or bl,al

int 21h
jmp input



exit:

lea dx,msg2
mov ah,9
```

```
int 21h

mov cx,4

convert:

mov dl,bh
shr dl,1
shr dl,1
shr dl,1
shr dl,1

cmp dl,9

jbe num
add dl,55d
jmp display

num: add dl,30h

display:

mov ah,2
```

```
int 21h


rcl bx,1
rcl bx,1
rcl bx,1
rcl bx,1


loop convert


main endp
end main



.MODEL SMALL
.STACK 100H

.DATA
PROMPT_1  DB  0DH,0AH,'Enter the first binary number ( max 8-digits ) : $'
PROMPT_2  DB  0DH,0AH,'Enter the second binary number ( max 8-digits ) : $'
PROMPT_3  DB  0DH,0AH,'The SUM of given binary numbers in binary form is : $'
ILLEGAL   DB  0DH,0AH,'Illegal character. Try again.$'

.CODE
```

```
MAIN PROC
MOV AX, @DATA              ; initialize DS
MOV DS, AX

JMP @START_2               ; jump to label @START_2

@START_1:                  ; jump label
LEA DX, ILLEGAL            ; load and display the string ILLEGAL
MOV AH, 9
INT 21H

@START_2:                  ; jump label
XOR BX, BX                 ; clear BX

LEA DX, PROMPT_1           ; load and display the string PROMPT_1
MOV AH, 9
INT 21H

MOV CX, 8                  ; initialize loop counter
MOV AH, 1                  ; set input function

@LOOP_1:                   ; loop label
INT 21H                    ; read a character

CMP AL, 0DH                ; compare AL with CR
```

```
JNE @SKIP_1          ; jump to label @SKIP_1 if AL!=0DH

CMP CX, 8            ; compare CX with 8
JE @START_1          ; jump to label @START_1 if CX=8
JMP @EXIT_LOOP_1     ; jump to label @EXIT_LOOP_1

@SKIP_1:             ; jump label
AND AL, 0FH          ; convert ascii into decimal code
SHL BL, 1            ; shift BL towards left by 1 position
OR BL, AL            ; set the LSB of BL with LASB of AL
LOOP @LOOP_1         ; jump to label @LOOP_1 if CX!=0

@EXIT_LOOP_1:        ; jump label

LEA DX, PROMPT_2     ; load and display the string PROMPT_2
MOV AH, 9
INT 21H

MOV CX, 8            ; initialize loop counter
MOV AH, 1            ; set input function

@LOOP_2:             ; loop label
INT 21H              ; read a character

CMP AL, 0DH          ; compare AL with CR
```

```
JNE @SKIP_2          ; jump to label @SKIP_2 if AL!=0DH

CMP CX, 8            ; compare CX with 8
JE @START_2          ; jump to label @START_2 if CX=8
JMP @EXIT_LOOP_2     ; jump to label @EXIT_LOOP_2

@SKIP_2:             ; jump label
AND AL, 0FH          ; convert ascii into decimal code
SHL BH, 1            ; shift BH towards left by 1 position
OR BH, AL            ; set the LSB of BH with LASB of AL
LOOP @LOOP_2         ; jump to label @LOOP_2 if CX!=0

@EXIT_LOOP_2:        ; jump label

LEA DX, PROMPT_3     ; load and display the string PROMPT_3
MOV AH, 9
INT 21H

ADD BL, BH           ; add BL and BH
JNC @SKIP            ; jump to label @SKIP if CF=1
MOV AH, 2            ; print the digit 1 i.e. carry
MOV DL, 31H
INT 21H

@SKIP:               ; jump label
```

```
MOV CX, 8              ; initialize loop counter
MOV AH, 2              ; set output function

@LOOP_3:               ; loop label
SHL BL, 1              ; shift BL towards left by 1 position
JC @ONE                ; jump to label @ONE if CF=1
MOV DL, 30H            ; set DL=0
JMP @DISPLAY           ; jump to label @DISPLAY

@ONE:                  ; jump label
MOV DL, 31H            ; set DL=1

@DISPLAY:              ; jump label
INT 21H                ; print the character
LOOP @LOOP_3           ; jump to label @LOOP_3 if CX!=0

MOV AH, 4CH            ; return control to DOS
INT 21H
MAIN ENDP
END MAIN
```

Ex:13

```
.MODEL SMALL
.STACK 100H

.DATA
PROMPT_1  DB  0DH,0AH,'Enter the first Hex number ( 0000 - FFFF ) : $'
PROMPT_2  DB  0DH,0AH,'Enter the second Hex number ( 0000 - FFFF ) : $'
PROMPT_3  DB  0DH,0AH,'The SUM of given Hex numbers in Hex form is : $'
ILLEGAL   DB  0DH,0AH,'Illegal character. Try again.$'

COUNT     DB  ?
VALUE     DW  ?

.CODE
MAIN PROC
MOV AX, @DATA              ; initialize DS
MOV DS, AX

JMP @START_2               ; jump to label @START_2
```

```
@START_1:              ; jump label
LEA DX, ILLEGAL        ; load and display the string ILLEGAL
MOV AH, 9
INT 21H

@START_2:              ; jump label
LEA DX, PROMPT_1       ; load and display the string PROMPT_1
MOV AH, 9
INT 21H

XOR BX, BX             ; clear BX
MOV COUNT, 30H         ; initialize loop counter

@START_3:              ; jump label
MOV AH, 1              ; set input function
INT 21H                ; read a character

CMP AL, 0DH            ; compare AL with CR
JNE @SKIP_1            ; jump to label @SKIP_1 if AL!=CR

CMP COUNT, 30H         ; compare COUNT with 0
JBE @START_1           ; jump to label @START_1 if COUNT<=0
JMP @END_1             ; jump to label @END
```

```
@SKIP_1:              ; jump label

CMP AL, "A"           ; compare AL with "A"
JB @DECIMAL_1         ; jump to label @DECIMAL_1 if AL<A

CMP AL, "F"           ; compare AL with "F"
JA @START_1           ; jump to label @START_1 if AL>F
ADD AL, 09H           ; add 9 to AL
JMP @OK_1             ; jump to label @OK_1

@DECIMAL_1:           ; jump label
CMP AL, 30H           ; compare AL with 0
JB @START_1           ; jump to label @START_1 if AL<0

CMP AL, 39H           ; compare AL with 9
JA @START_1           ; jump to label @START_1 if AL>9

@OK_1:                ; jump label

INC COUNT             ; increment the COUNT variable

AND AL, 0FH           ; convert the ascii into binary code

MOV CL, 4             ; set CL=4
SHL AL, CL            ; shift AL towards left by 4 positions
```

```
MOV CX, 4              ; set CX=4

@LOOP_1:               ; loop label
SHL AL, 1              ; shift AL towards left by 1 position
RCL BX, 1              ; rotate BX towards left by 1 position
; through carry
LOOP @LOOP_1           ; jump to label @LOOP_1 if CX!=0

CMP COUNT, 34H         ; compare COUNT with 4
JE @END_1              ; jump to label @END_1 if COUNT=4
JMP @START_3           ; jump to label @START_3

@END_1:                ; jump label

MOV VALUE, BX          ; set VALUE=BX

LEA DX, PROMPT_2       ; load and display the string PROMPT_2
MOV AH, 9
INT 21H

XOR BX, BX             ; clear BX
MOV COUNT, 30H         ; initialize loop counter

@START_4:              ; jump label
```

```
MOV AH, 1            ; set input function
INT 21H              ; read a character

CMP AL, 0DH          ; compare AL with CR
JNE @SKIP_2          ; jump to label @SKIP_2 if AL!=CR

CMP COUNT, 30H       ; compare COUNT with 0
JBE @START_1         ; jump to label @START_1 if COUNT<=0
JMP @END_2           ; jump to label @END_2

@SKIP_2:             ; jump label

CMP AL, "A"          ; compare AL with "A"
JB @DECIMAL_2        ; jump to label @DECIMAL_2 if AL<A

CMP AL, "F"          ; compare AL with "F"
JA @JUMP             ; jump to label @JUMP if AL>F
ADD AL, 09H          ; add 9 to AL
JMP @OK_2            ; jump to label @OK_2

@DECIMAL_2:          ; jump label
CMP AL, 30H          ; compare AL with 0
JB @JUMP             ; jump to label @JUMP if AL<0

CMP AL, 39H          ; compare AL with 9
```

```
JA @JUMP            ; jump to label @JUMP if AL>9
JMP @OK_2           ; jump to label @OK_2

@JUMP:              ; jump label
JMP @START_1        ; jump to label @START_1

@OK_2:              ; jump label

INC COUNT           ; increment the COUNT variable

AND AL, 0FH         ; convert the ascii into binary code

MOV CL, 4           ; set CL=4
SHL AL, CL          ; shift AL towards left by 4 positions

MOV CX, 4           ; set CX=4

@LOOP_2:            ; loop label
SHL AL, 1           ; shift AL towards left by 1 position
RCL BX, 1           ; rotate BX towards left by 1 position
; through carry
LOOP @LOOP_2        ; jump to label @LOOP_2 if CX!=0

CMP COUNT, 34H      ; compare COUNT with 4
JE @END_2           ; jump to label @END_2 if COUNT=4
```

```
JMP @START_4          ; jump to label @START_4

@END_2:               ; jump label

LEA DX, PROMPT_3      ; load and display the string PROMPT_3
MOV AH, 9
INT 21H

ADD BX, VALUE         ; add BX and VALUE
JNC @SKIP             ; jump to label @SKIP if CF=1
MOV AH, 2             ; set output function
MOV DL, 31H           ; set DL=1
INT 21H               ; print a character

@SKIP:                ; jump label

MOV COUNT, 30H        ; set COUNT=0

@LOOP_3:              ; loop label
XOR DL, DL            ; clear DL
MOV CX, 4             ; set CX=4

@LOOP_4:              ; loop label
SHL BX, 1             ; shift BX towards left by 1 position
RCL DL, 1             ; rotate DL towards left by 1 position
```

```
; through carry
LOOP @LOOP_4            ; jump to label @LOOP_4 if CX!=0

MOV AH, 2               ; set output function

CMP DL, 9               ; compare DL with 9
JBE @NUMERIC_DIGIT      ; jump to label @NUMERIC_DIGIT if DL<=9
SUB DL, 9               ; convert it to number i.e. 1,2,3,4,5,6
OR DL, 40H              ; convert number to letter i.e. A,B...F
JMP @DISPLAY            ; jump to label @DISPLAY

@NUMERIC_DIGIT:         ; jump label
OR DL, 30H              ; convert decimal to ascii code

@DISPLAY:               ; jump label
INT 21H                 ; print the character

INC COUNT               ; increment COUNT variable
CMP COUNT, 34H          ; compare COUNT with 4
JNE @LOOP_3             ; jump to label @LOOP_3 if COUNT!=4

@END:                   ; jump label

MOV AH, 4CH             ; return control to DOS
INT 21H
```

MAIN ENDP

END MAIN

Ex:14

```
.MODEL SMALL
.STACK 100H

.DATA
PROMPT_1  DB  'Enter a decimal digit string : $'
PROMPT_2  DB  0DH,0AH,'The sum of the decimal digit string in Hex is : $'
ILLEGAL   DB  0DH,0AH,'Illegal character. Try again : $'

TEMP      DW  ?
VALUE     DW  ?
v         dw  ?

.CODE
MAIN PROC
MOV AX, @DATA              ; initialize DS
MOV DS, AX

LEA DX, PROMPT_1           ; load and display the string PROMPT_1
MOV AH, 9
INT 21H

JMP @START_2               ; jump to label @START_2

@START_1:                  ; jump label
LEA DX, ILLEGAL            ; load and display the string ILLEGAL
```

```
MOV AH, 9
INT 21H

XOR BX, BX              ; clear BX
XOR CX, CX              ; clear CX

@START_2:               ; jump label
MOV AH, 1
INT 21H

INC CX                  ; increment CX

CMP AL, 0DH             ; compare AL with CR
JNE @SKIP               ; jump to label @SKIP if AL!=CR

CMP CX, 1               ; compare CX with 1
JB @START_1             ; jump to label @START_1 if CX<1
JMP @END_INPUT          ; jump to label @END_INPUT

@SKIP:                  ; jump label

CMP AL, 30H             ; compare AL with 0
JB @START_1             ; jump to label @START_1 of AL<0

CMP AL, 39H             ; compare AL with 1
```

```
JA @START_1            ; jump to label @START_1 if AL>9

AND AL, 0FH            ; convert the ascii to decimal code
XOR AH, AH             ; clear AH
ADD BX, AX             ; add BX and AX

JMP @START_2           ; jump to label @START_2

@END_INPUT:            ; jump label

LEA DX, PROMPT_2       ; load and display the string PROMPT_2
MOV AH, 9
INT 21H

MOV CX, 4              ; initialize loop counter
MOV AH, 2              ; set output function

@LOOP_1:               ; loop label
XOR DX, DX             ; clear DX

@LOOP_2:               ; jump label
SHL BX, 1              ; shift BX towards left by 1 position
RCL DL, 1              ; rotate DL towards left by 1 position
INC DH                 ; increment DH
CMP DH, 4              ; compare DH with 4
```

```
JNE @LOOP_2             ; jump to label @LOOP_2 if DH!=4


CMP DL, 9               ; compare DL with 9
JBE @NUMERIC_DIGIT      ; jump to label @NUMERIC_DIGIT if DL<=9
SUB DL, 9               ; convert it to number i.e. 1,2,3,4,5,6
OR DL, 40H              ; convert number to letter i.e. A,B...F
JMP @DISPLAY            ; jump to label @DISPLAY

@NUMERIC_DIGIT:         ; jump label
OR DL, 30H              ; convert decimal to ascii code

@DISPLAY:               ; jump label
INT 21H                 ; print the character
LOOP @LOOP_1            ; jump to label @LOOP_1 if CX!=0

MOV AH, 4CH             ; return control to DOS
INT 21H
MAIn ENDP
END MAIN
```