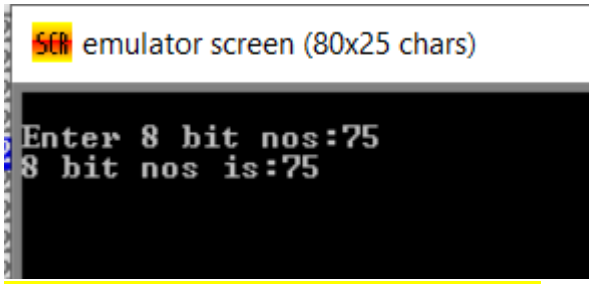
Experiment No 1
Apply Assembly Language Programing to enter and display 8 bit & 16 bits number

Program 1: Enter and display 8 bit no

```
.model small
.data
msg1 db 10,13,"Enter 8 bit nos :$"
msg2 db 10,13,"8 bit nos is :$"
.code
.startup
mov ah,09h
lea dx,msg1
int 21h

mov ah,01h
int 21h
sub al,30h
mov cl,04h
shl al,cl

mov bl,al
mov ah,01h
int 21h
sub al,30h
add al,bl
mov bh,al
mov ah,09h
lea dx,msg2
int 21h

mov bl,bh
and bl,0f0h
shr bl,cl
add bl,30h
mov dl,bl
mov ah,02h
int 21h

mov bl,bh
and bl,0fh
add bl,30h
mov dl,bl
mov ah,02h
int 21h

.exit
end
```



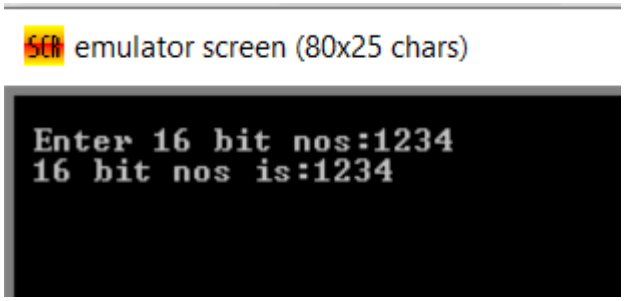
Program 2: Enter and display 16 bit no

```
.model small
.data
msg1 dw 10,13,"Enter 16 bit nos :$"
msg2 dw 10,13,"16 bit nos is :$"
.code
.startup
mov ah,09h
lea dx,msg1
int 21h

mov ah,01h     ;input 1st digit
int 21h
sub al,30h
mov cl,04h
shl al,cl
mov bh,al

mov ah,01h    ;input 2nd digit
int 21h
sub al,30h
add bh,al


mov ah,01h  ;input 3rd digit
int 21h
sub al,30h
mov cl,04h
shl al,cl
mov bl,al

mov ah,01h   ;input 4th digit
int 21h
sub al,30h
add bl,al
```

;Display 16 bit no

```
mov ah,09h
lea dx,msg2
int 21h
```

```
mov ch,bh
and ch,0f0h
mov cl,04h
shr ch,cl
add ch,30h
mov dl,ch
mov ah,02h
int 21h

mov ch,bh
and ch,0fh
add ch,30h
mov dl,ch
mov ah,02h
int 21h

mov dh,bl
and dh,0f0h
mov cl,04h
shr dh,cl
add dh,30h
mov dl,dh
mov ah,02h
int 21h

mov dh,bl
and dh,0fh
add dh,30h
mov dl,dh
mov ah,02h
int 21h
.exit
end
```

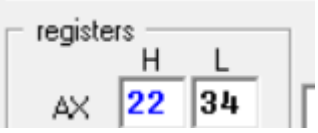


Experiment No 2

Apply Assembly Language Programing to perform addition and subtraction of two 16 bits numbers using macros and procedure.

Addition of two 16 bit nos using procedure

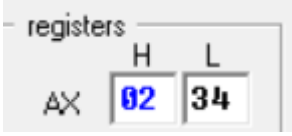
```
.model small
.data
num1 dw 1234H
num2 dw 1000H
res dw ?
.code
mov ax,@data
mov ds,ax
call addproc

mov ah,4ch
int 21H

proc addproc
mov ax,num1
add ax,num2
mov res,ax
ret
endp

ends
End
```



Subtraction of two 16 bit nos using procedure

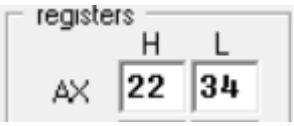
```
.model small
.data
num1 dw 1234H
num2 dw 1000H
res dw ?
.code
mov ax,@data
mov ds,ax
call subproc
mov ah,4ch
int 21H

proc subproc
mov ax,num1
```

```
sub ax,num2
mov res,ax
ret
endp

ends
End
```



Addition of two 16 bit nos using Macro

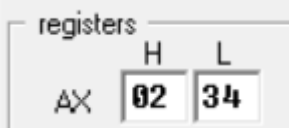
```
addm macro num1,num2
mov ax,num1
add ax,num2
mov res,ax
endm

.model small
.data
num1 dw 1234H
num2 dw 1000H
res dw ?
.code
mov ax,@data
mov ds,ax
addm num1,num2
mov ah,4ch
int 21H
ends
End
```



Subtraction of two 16 bit nos using Macro

```
subm macro num1,num2
mov ax,num1
sub ax,num2
mov res,ax
endm

.model small
.data
num1 dw 1234H
num2 dw 1000H
res dw ?
.code
mov ax,@data
```

```
mov ds,ax
subm num1,num2
mov ah,4ch
int 21H
ends
End
```



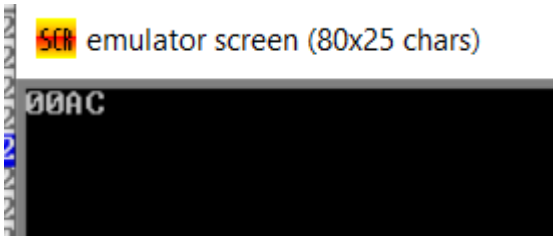
Experiment No 3

Apply Assembly Language Programing to covert HEX to BCD and BCD to HEX.

A. BCD to Hex Conversion

```
;bcd to hex

.model small
.data
Var DW 0172d
.code
;Initlize Data Segment
mov ax,@DATA
mov DS,ax
mov bx,Var
mov al,bh
mov ah,00h
mov cl,10h
div cl
mov dh,ah
mov cl,al
CALL PRINT
mov dl,dh
mov cl,dl
CALL PRINT
mov al,bl
mov ah,00h
mov cl,10h
div cl
mov dh,ah
mov cl,al
CALL PRINT
mov cl,dh
CALL PRINT
;to terminate program
mov ah, 4ch
int 21h
ret
```

```
PRINT PROC ;print procedure to print
;number
cmp cl,09
jle ad
add cl,07h ;if less than 9 ,add 30h
ad: add cl,30h ;if greater than 9,add 37h
mov dl,cl
mov ah,02h
int 21h
ret
PRINT ENDP

Ends
End
```



B. HEX to BCD Conversion

```
;hex to bcd
.model small
.data
hex dw 0ACH
counter db 0
.code
;initialize Data Segment
mov ax,@DATA
mov DS,ax
mov ax,hex
mov bx,000Ah

L:
inc counter
div bx
push dx
cmp ax,0
mov dx,00h
je exit
jmp L

exit:
mov cl,counter
mov ch,00h

L1:
```

```
pop dx
add dl,30h
mov ah,02h
int 21h

LOOP L1
;to terminate program
mov ah,4ch
int 21h
ret

ends
end
```

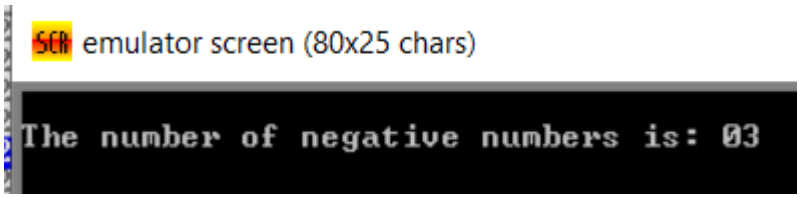
emulator screen (80x25 chars)

Experiment No 4
Finding Negative Numbers from given array

```
print macro m
mov ah,09h
mov dx,offset m
int 21h
endm



.model small

.data

list db 10,20,80h,86h,23,26,12,57,89h  ;array of numbers
count db (0)                           ;count variable(to store answer)

msg db 10,13, "The number of negative numbers is: $"  ;output message

.code

start:  mov ax,@data
        mov ds,ax

        mov ch,00                  ; temp storage of ans
        mov si,offset list         ;point to start of array
        mov cl,09                  ;count of numbers in the array

again:  mov al,[si]                ;copy num in al
        and al,80h                 ;AND with 80H
        jz next                    ; jump to next statement if result is zero
                                   ; i.e. positive number. Else increase coun
t
```

```
        inc ch                    ;increment count if negative number if AND
ing
                                  ; gives non zero value


next:

        inc si                   ;inc si to point to next location in array
        dec cl                   ;decrement count of the array to check
        jnz again                ;if all numbers aare not covered do again


        mov bl,ch                ;store the answer in bl



     ;printing the result

        print msg              ;print the string

        mov cl,04              ;count for shifting to display a number
        mov al,bl              ;copy ans in "AL" register

        and al,0f0h            ;Mask the LSB and take only MSB
        shr al,cl              ;shift the numberto bring MSB to LSB

        cmp al,09              ;if it is number 0-9 just add 30H
        jbe alpha              ; if character A-F add additional 7
        add al,07              ; for correct ASCII value to display


alpha:  add al,30h             ; add 30H to make the number ASCII
        mov ah,02              ; display function
        mov dl,al              ; content to be displyed needs to be in DL
                               ;  for 02 function
        int 21h


       ;printing LSB digit

        mov ah,02           ;02 function for single digit display
        mov al,bl           ;copy ans in Al register
        and al,00fh         ; Mask the MSB. Since number is in LSB no need t
o
                            ; shift

        cmp al,09           ; check if number in 0-9
        jbe alpha2          ; if alphabet add additional 7 to make correct
        add al,07           ;  ASCII value

alpha2:  add al,30h         ; add 30H for ASCII value
         mov dl,al          ; display content in DL for 02 function
         int 21h

         mov ah,4ch         ; 4ch function to terminate program and return
```

```
        int 21h             ;  to DOS prompt

        end start
        end                 ; end of file
```



Experiment No 5

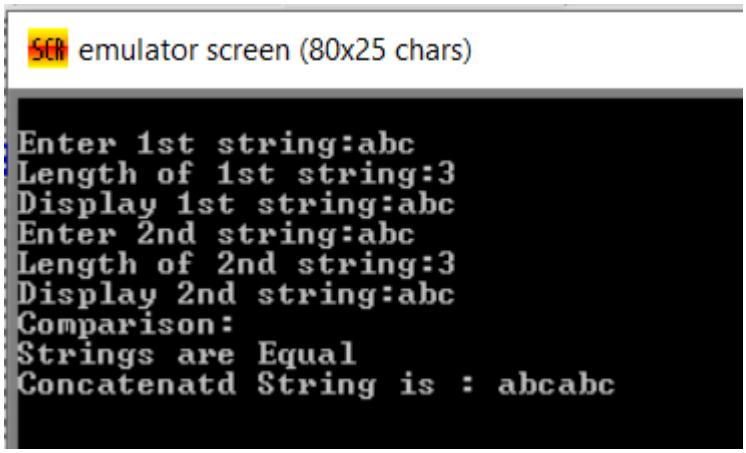
Student should be able to apply string operations (i) Accept, (ii) Display, (iii) Concatenate and (iv) Compare in ALP.

```
.model small
.stack
.data
m1 db 10,13,"Enter 1st string:$"
m2 db 10,13,"Length of 1st string:$"
m3 db 10,13,"Display 1st string:$"
m4 db 10,13,"Enter 2nd string:$"
m5 db 10,13,"Length of 2nd string:$"
m6 db 10,13,"Display 2nd string:$"
m7 db 10,13,"Comparison : $ "
m8 db 10,13,"Strings are Equal$"
m9 db 10,13," Strings are not Equal$"
m10 db 10,13,"Concatenatd String is  : $"

str1 db 80,?,80 DUP(?)
str2 db 80,?,80 DUP(?)
str3 db 80,?,80 DUP(?)

.code
Disp macro xx
   mov ah,09h
   lea dx,xx
   int 21h
endm

.startup
Disp m1     ;Enter 1st string
mov ah,0Ah  ;Read a string from the keyboard into buffer addressed by DX
lea dx,str1
int 21h

Disp m2       ;Length of 1st string
```

```
lea si,str1+1
mov dl,[si]
mov cl,dl
add dl,30h
mov ah,02h
int 21h

Disp m3       ;Display 1st string
lea si,str1+2

Back:
mov dl,[si]
mov ah,02h
int 21h

inc si
dec cl
jnz Back

Disp m4       ;Enter 2nd string
mov ah,0Ah
lea dx,str2
int 21h

Disp m5    ;Length of 2nd string
lea si,str2+1
mov dl,[si]
mov cl,dl
add dl,30h
mov ah,02h
int 21h

Disp m6    ;Display 2nd string
lea si,str2+2
Back1:
mov dl,[si]
mov ah,02h
int 21h

inc si
dec cl
jnz Back1

Disp m7      ; Comparison
lea si,str1+1
mov cl,[si]
lea di,str2+1
mov ch,[di]
cmp cl,ch
jnz AA
lea si,str1+2
```

```
lea di,str2+2

Back2:
mov dl,[si]
mov dh,[di]
cmp dl,dh
jnz AA
inc si
inc di
dec cl
jnz Back2

Disp m8    ;Strings are Equal
jmp con

AA:
Disp m9   ; Strings are not Equal

con:
Disp m10       ;Concatenatd String is

lea si,str1+1
mov cl,[si]
mov bl,cl

lea di,str2+1
mov ch,[di]
mov bh,ch
add bl,bh

lea si,str1+2
lea di,str3+2

Back3:
mov dl,[si]
mov [di],dl
inc si
inc di
dec cl
jnz back3
lea si,str2+2

Back4:
mov dl,[si]
mov [di],dl
inc si
inc di
dec ch
jnz Back4
lea di,str3+2
```

```
Back5:
mov dl,[di]
mov ah,02h
int 21h
inc di
dec bl
jnz Back5

Exit:
.exit
end
```



Experiment No 6

Write a mixed language code for designing calculator

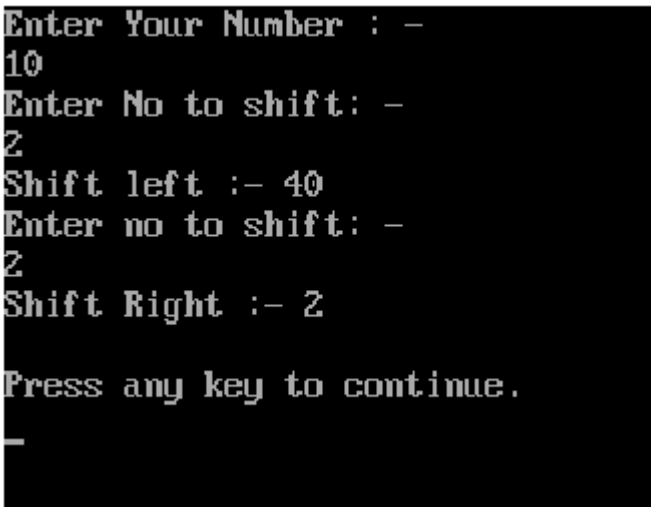
```
#include <stdio.h>
void main() {
int a, b, c;
printf("\n\n");
printf(" Enter first number a: ");
scanf("%d",&a);
printf(" Enter second number b: ");
scanf("%d",&b);
printf("\n a=%d",a);
printf("\n b=%d",b);
asm {
mov ax,a
mov bx,b
add ax,bx
mov c,ax
}
printf("\n\n The addition of a and b is
%d",c);
asm {
mov ax,a
mov bx,b
sub ax,bx
```

```
mov c,ax
}
printf("\n\n The subtraction of a and b
is %d",c);
asm {
mov ax,a
mov bx,b
mul bx
mov c,ax
}
printf("\n\n The multiplication of a and
b is %d",c);
asm {
mov ax,a
mov bx,b
div bx
mov c,ax
}
printf("\n\n The division of a and b is
%d",c);
printf("\n\n");
}
```

```
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
main.c:
Turbo Link  Version 5.0 Copyright (c) 1992 Borland International

        Available memory 4140944


 Enter first number a: 8
 Enter second number b: 3

 a=8
 b=3

 The addition of a and b is 11

 The subtraction of a and b is 5

 The multiplication of a and b is 24

 The division of a and b is 2


Press any key to continue.
```

```
#include <iostream.h>
#include <conio.h>

int main()
{
   clrscr();
   int a, b, c, d;
```

```
    cout << "Enter Your Number : -  " << endl;
    cin >> a;

    cout << "Enter No to shift: - " << endl;
    cin >> b;

    asm mov ax, a;
    asm mov cx, b;
    asm shl ax, cl;
    asm mov c, ax;
    cout << "Shift left :- " << c << endl;

    cout << "Enter no to shift: - " << endl;
    cin >> d;
    asm mov ax, a;
    asm mov cx, d;
    asm shr ax, cl;
    asm mov c, ax;

    cout << "Shift Right :- " << c << endl;
    getch();
    return 0;
}
```



Experiment No 7 : Interfacing of Mouse Driver

```
.model small
.stack
.data
msg1 db 10,13, "Mouse driver present:"
.code
disp macro xx
mov ah,09
lea dx,xx
```

```
int 21h
endm

.startup
mov ax,0000   ;mouse driver check
int 33h

cmp ax,00h
je last
disp msg1

mov ax,0004  ;mouse cursor position
mov cx,0
mov dx,0
int 33h

mov ax, 0007 ;set horizontal limit
mov cx,0010
mov dx,055h
int 33h

mov ax, 0008 ;set vertical limit
mov cx,0010
mov dx,055h
int 33h

pixel:
mov ax,0001 ;display mouse cursor
int 33h

mov ax,0003
int 33h
cmp bx,01 ;left button
je left
jmp right

left:
mov bx,0011h   ;set graphics mode
int 10h

mov ah,0ch ;display pixel on screen
int 10h

right:
mov ax,0001
int 33h
cmp bx,02
je last
jmp pixel

last:
```

```
mov ax,00 ;set text mode
int 10h

.exit
end
```