System Programming Lab
BCSE 3rd year 1st Semester
Name: Sayantan Biswas
Roll No: 001910501057

>> I have done this assignment using emu8086

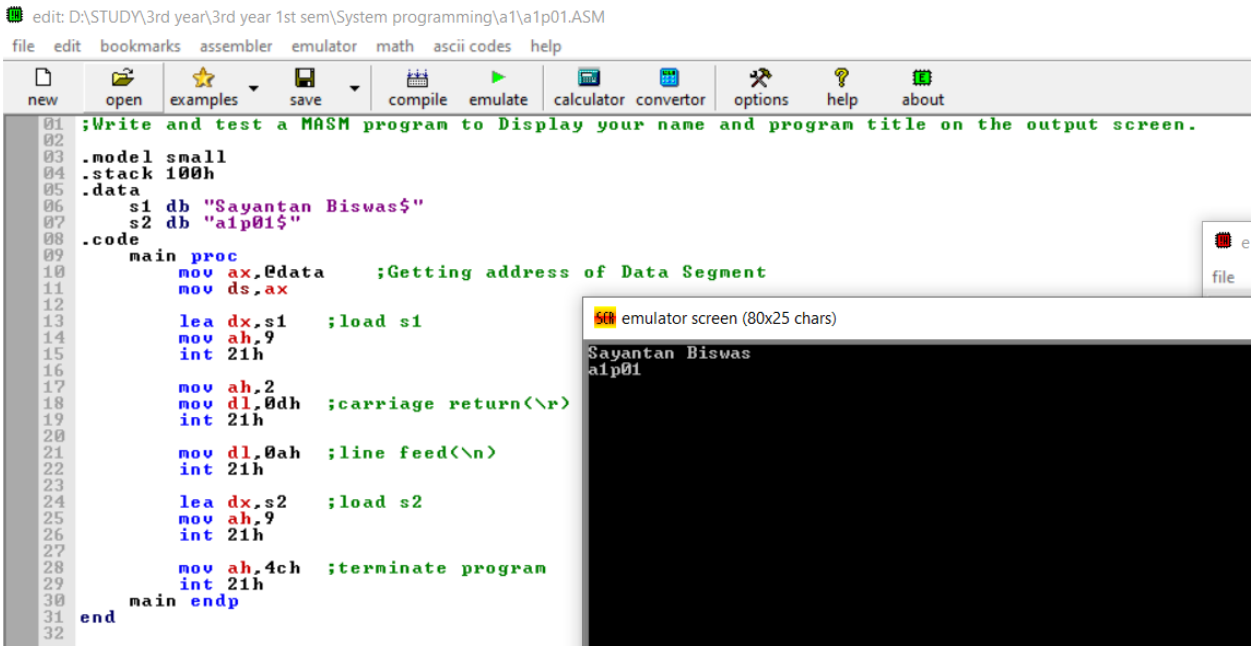
Assignment 1

Q1:

;Write and test a MASM program to Display your name and program title on the output screen.

```
.model small

.stack 100h

.data

   s1 db "Sayantan Biswas$"

   s2 db "a1p01$"

.code

   main proc

      mov ax,@data    ;Getting address of Data Segment

      mov ds,ax


      lea dx,s1   ;load s1
```

```
        mov ah,9
        int 21h

        mov ah,2
        mov dl,0dh  ;carriage return(\r)
        int 21h

        mov dl,0ah  ;line feed(\n)
        int 21h

        lea dx,s2   ;load s2
        mov ah,9
        int 21h

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

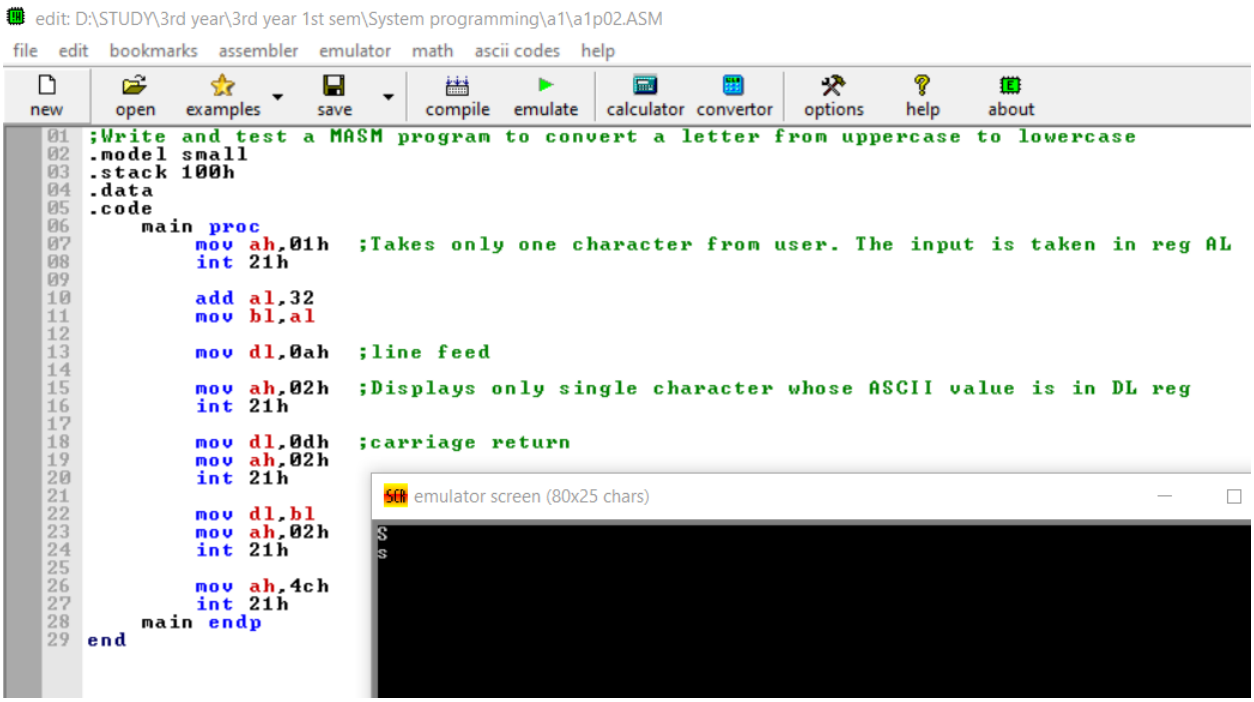


Q2:

;Write and test a MASM program to convert a letter from uppercase to lowercase

```
.model small

.stack 100h

.data

.code

   main proc

      mov ah,01h  ;Takes only one character from user. The input is taken in reg AL

      int 21h


      add al,32

      mov bl,al
```

```
        mov dl,0ah  ;line feed

        mov ah,02h  ;Displays only single character whose ASCII value is in DL reg
        int 21h

        mov dl,0dh  ;carriage return
        mov ah,02h
        int 21h

        mov dl,bl
        mov ah,02h     ;Displays only single character whose ASCII value is in DL reg
        int 21h

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



Q3:

;Write and test a MASM program to add two Hexadecimal Numbers

```
.MODEL SMALL

.STACK 100H

.DATA

   L1 db 13,10,"Enter the 1st number: $"

   L2 db 13,10,"Enter the 2nd number: $"

   L3 db 13,10,"The result after addition is: $"


.CODE

  MAIN PROC

        MOV AX,@DATA
```

```
    MOV DS,AX

    XOR BX,BX      ;set BX value to 0
    MOV CL,4

    LEA DX, L1     ;show L1
    MOV AH, 9
    INT 21H

    MOV AH,1       ;for taking input
    INT 21H

  INPUT1:
    CMP AL,0DH     ;compare whether the pressed key is 'ENTER' or not
    JE LINE1       ;If it is equal to 'Enter' then stop taking first value. JE = Jump if equal

    CMP AL,39H     ;compare the input whether it is letter or digit. 39h is the ascii value of 9
    JG LETTER1     ;JG = Jump if greater/not less or equal
```

```
    AND AL,0FH     ;if it is digit then convert it's ascii value to real value, clear the high-order bits to zero
    JMP SHIFT1

  LETTER1:         ;if it is letter then subtract 37h from it to find it's real value. 41h is the ASCII code for a letter 'A', (41h - 10d) = (41h - 0Ah) = 37h
    SUB AL,37h

  SHIFT1:
    SHL BX, CL     ;shift the bits left
    OR  BL,AL      ;making 'or' to add the current value with previous value

    INT 21H
    JMP INPUT1

  LINE1:
    LEA DX, L2     ;show L2
    MOV AH, 9
    INT 21H

    XOR DX,DX      ;set DX value zero
```

```
    MOV AH,1
    INT 21H


INPUT2:
    CMP AL,0DH      ;compare whether the pressed key is 'ENTER' or not
    JE LINE2        ;If it is equal to 'Enter' then stop taking first value. JE = Jump if equal

    CMP AL,39H      ;compare the input whether it is letter or digit. 39h is the ascii value of 9
    JG LETTER2      ;JG = Jump if greater/not less or equal

    AND AL,0FH      ;if it is digit then convert it's ascii value to real value, clear the high-order bits to zero
    JMP SHIFT2

LETTER2:            ;if it is letter then subtract 37h from it to find it's real value. 41h is the ASCII code for a letter 'A', (41h - 10d) = (41h - 0Ah) = 37h
    SUB AL,37H

SHIFT2:
    SHL DX, CL
```

```
    OR  DL,AL       ;making 'or' to add the current value with previous value


    INT 21H
    JMP INPUT2
LINE2:
    XOR CX,CX
    MOV CX,DX


    MOV DH,16


SUM:
    ADD BX,CX       ;add two number which are stored in BX and CX register
    JC PC1          ;if the register is overflowed then print an extra 1


    mov cl, 4


    LEA DX, L3      ;show L3
    MOV AH, 9
    INT 21H


OUTPUT:             ;for printing the sum
```

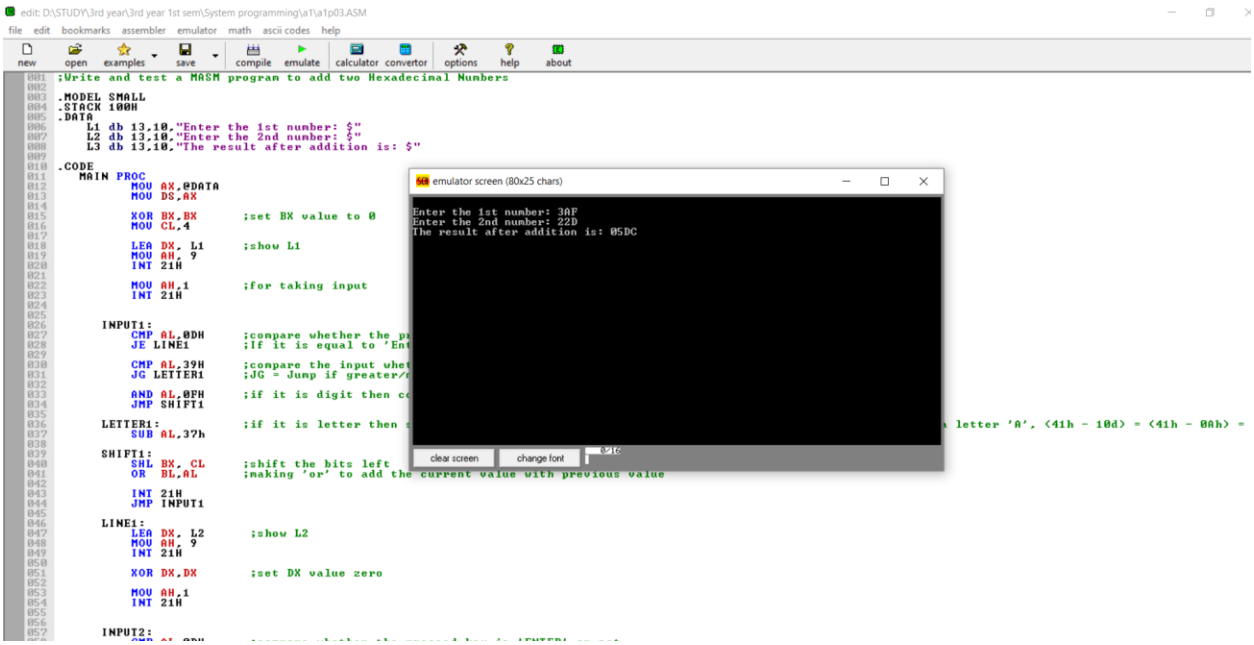
```
    MOV CH,BH
    SHR CH, CL
    AND CH,0FH

    CMP CH,10       ;convert decimal to binary
    ADD CH,48
    CMP CH,':'
    JL TAG          ;Jump to Label
    ADD CH,7

TAG:                ; 4 Tags for 4 digit display
    MOV DL,CH
    MOV AH,2
    INT 21H

    MOV CH,BH
    AND CH,0FH
    CMP CH,10
    ADD CH,48
    CMP CH,':'
    JL TAG1
    ADD CH,7
```

```
TAG1:
    MOV DL,CH
    MOV AH,2
    INT 21H

    MOV CH,BL
    SHR CH, CL
    AND CH,0FH
    CMP CH,10
    ADD CH,48
    CMP CH,':'
    JL TAG2
    ADD CH,7

TAG2:
    MOV DL,CH
    MOV AH,2
    INT 21H

    MOV CH,BL
    AND CH,0FH
```

```
    CMP CH,10
    ADD CH,48
    CMP CH,':'
    JL TAG3
    ADD CH,7

TAG3:
    MOV DL,CH
    MOV AH,2
    INT 21H

    JMP EXIT

PC1:            ;for printing overflowed 1
    MOV DL,'1'
    MOV AH,2
    INT 21H
    JMP OUTPUT

EXIT:
    MOV AH, 4CH
    INT 21H
```

MAIN ENDP

END MAIN



Q4:

;Write and test a MASM program to find the second max and second min from an array.

.model small

.stack 100h

.data

prompt_0 db 'enter the number of array elements :',0dh,0ah,'$'

prompt_1 db 'enter the array elements :',0dh,0ah,'$'

```
prompt_2  db  'the 2nd maximum is : $'
prompt_3  db  'the 2nd minimum is : $'

array   dw  50 dup(0)

s dw ?
max dw ?
min dw ?

.code
main proc

                mov ax, @data              ; initialize ds
                mov ds, ax

                lea dx, prompt_0           ; load and display the string prompt_0
                mov ah, 9
                int 21h

                mov ah,1                           ;for taking input
                int 21h
```

```
input1:
cmp al,0dh                    ;compare whether the pressed key is
'enter' or not
je line1                      ;if it is equal to 'enter' then stop taking
first value

and al,0fh                    ;convert it's ascii value to real value by
masking

shl bx, 1
shl bx, 1
shl bx, 1
shl bx, 1
or  bl,al                     ;making 'or' will add the current value
with previous value

int 21h
jmp input1

line1:
lea dx, prompt_1          ; load and display the string prompt_1
mov ah, 9
int 21h
```

```
lea si, array          ; set si=offset address of array
mov s,bx
mov cx, bx             ; set cx=bx

@read_array:           ; loop label

mov ah,1               ;for taking input
int 21h

xor dx,dx

input2:
cmp al,0dh             ;compare whether the pressed key is 'enter' or not
je line2               ;if it is equal to 'enter' then stop taking first value

and al,0fh             ;convert it's ascii value to real value by masking

shl dx,1
```

```
shl dx,1

shl dx,1

shl dx,1

or  dl,al                    ;making 'or' will add the current value
with previous value


int 21h

jmp input2


line2:

mov [si], dx              ; set [si]=ax

add si, 2                 ; set si=si+2


mov dl, 0ah               ; line feed

mov ah, 2                 ; set output function

int 21h                   ; print a character


loop @read_array          ; jump to label @read_array while cx!=0

; array input done


lea si,array

mov ax,bx
```

```
dec ax
xor bx,bx
xor cx,cx
mov bx,word ptr[si]        ;store the maximum
mov cx,word ptr[si]        ;store the 2nd
add si, 2

; loop to find max and 2nd max
arrayloop2:

cmp word ptr[si],bx
jl max2
mov cx,bx
mov bx,word ptr[si]

max2:
cmp word ptr[si],cx
jl incre
cmp word ptr[si],bx
je incre
mov cx,word ptr[si]
```

```
incre:
add si, 2
dec ax

jnz arrayloop2

; now bx has max cx has 2nd max
mov max,bx
; displaying the prompt
lea dx,prompt_2
mov ah,09h
int 21h

; display contents of cx
mov bx,cx

mov dh,bh
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh
```

```
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bh
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
```

```
int 21h

mov dh,bl
and dh,0fh
cmp dh,10
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dl, 0ah              ; line feed
mov ah, 2                ; set output function
int 21h                  ; print a character

lea si,array
mov ax,s
dec ax
mov bx,max


; loop to find min and 2nd min
arrayloop3:
```

```
cmp word ptr[si],bx
jg min2
mov cx,bx
mov bx,word ptr[si]

min2:
cmp word ptr[si],cx
jg incre2
cmp word ptr[si],bx
je incre2
mov cx,word ptr[si]

incre2:
add si, 2
dec ax

jnz arrayloop3

; now bx has min cx has 2nd min

; displaying the prompt
```

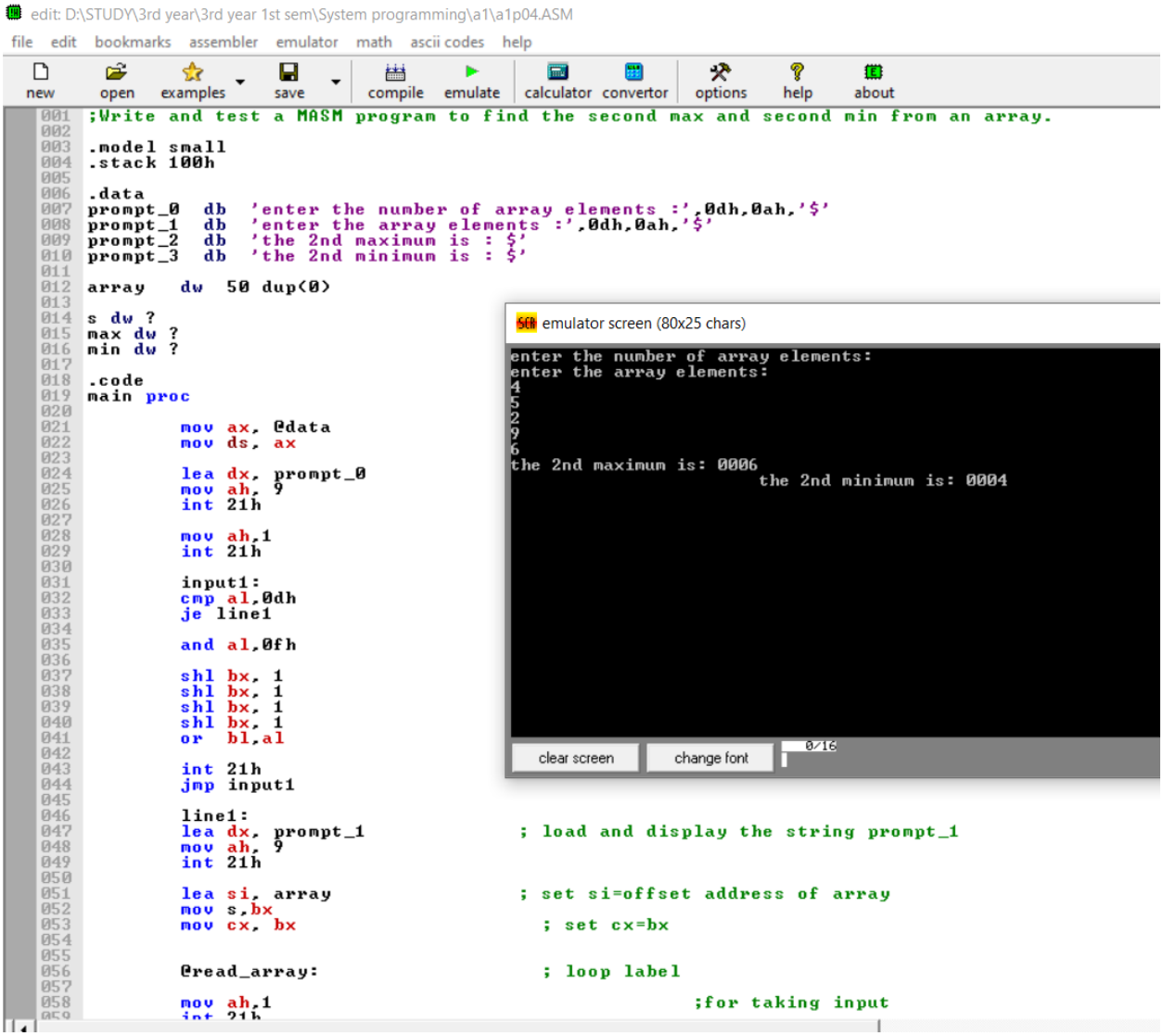
```
lea dx,prompt_3
mov ah,09h
int 21h

; display contents of cx
mov bx,cx

mov dh,bh
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh

add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bh
and dh,0fh
add dh,'0'
```

```
mov dl,dh
mov ah,2
int 21h

mov dh,bl
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
and dh,0fh
cmp dh,10
add dh,'0'
mov dl,dh
mov ah,2
int 21h
```

```
exit:
mov ah, 4ch                    ;return control to dos
int 21h
```

main endp

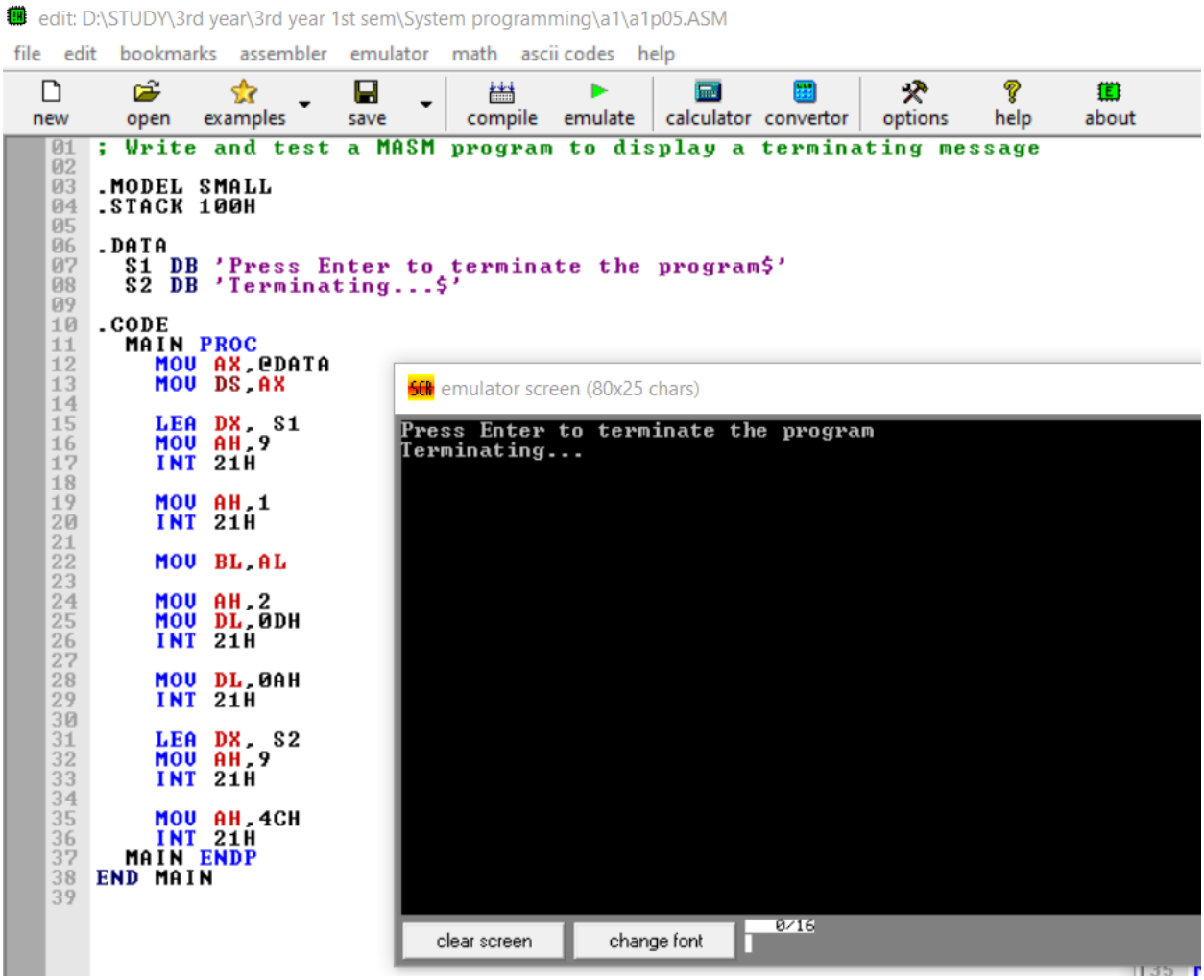
end main



Q5:

```
; Write and test a MASM program to display a terminating message


.MODEL SMALL

.STACK 100H


.DATA

 S1 DB 'Press Enter to terminate the program$'

 S2 DB 'Terminating...$'


.CODE

 MAIN PROC

  MOV AX,@DATA

  MOV DS,AX


  LEA DX, S1

  MOV AH,9          ; Displays a message terminated by “$”

  INT 21H


  MOV AH,1        ;Takes only one character from user. The input is taken in reg AL

  INT 21H
```

```
    MOV BL,AL

    MOV AH,2        ;cursor position
    MOV DL,0DH      ;carriage return
    INT 21H

    MOV DL,0AH      ;line feed
    INT 21H

    LEA DX, S2
    MOV AH,9
    INT 21H

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



Q6:

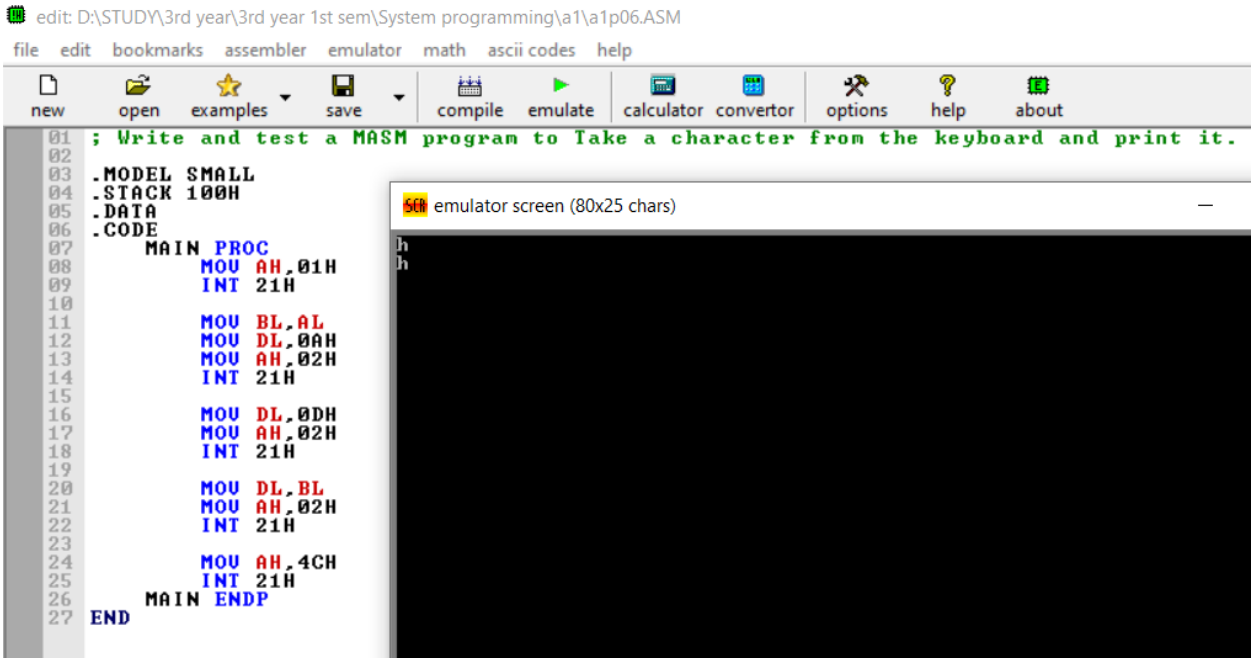
; Write and test a MASM program to Take a character from the keyboard and print it.

.MODEL SMALL

.STACK 100H

```
.DATA
.CODE
  MAIN PROC
    MOV AH,01H  ;Takes only one character from user
    INT 21H

    MOV BL,AL
    MOV DL,0AH
    MOV AH,02H
    INT 21H

    MOV DL,0DH
    MOV AH,02H
    INT 21H

    MOV DL,BL
    MOV AH,02H
    INT 21H

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

END



Q7:

; Write and test a MASM program to validate second numbers is less than the first.

.MODEL SMALL

.STACK 100H

.DATA

```
    S1 DB 0AH,0DH,"Enter 1st number: $"
    S2 DB 0AH,0DH,"Enter 2nd number: $"
    S3 DB 0AH,0DH,"2nd number is less than 1st number$"
    S4 DB 0AH,0DH,"2nd number is greater than 1st number$"
    S5 DB 0AH,0DH,"Two numbers are equal$"
.CODE
    MAIN PROC
        MOV AX,@DATA
        MOV DS,AX

        LEA DX,S1
        MOV AH,09H
        INT 21H

        MOV AH,01H  ;takes one input
        INT 21H

        MOV BL,AL
        LEA DX,S2
        MOV AH,09H
        INT 21H
```

```
MOV AH,01H
INT 21H

CMP BL,AL
JG COND1
JL COND2
LEA DX,S5
MOV AH,09H
INT 21H

MOV AH,4CH
INT 21H

COND1:
   LEA DX,S3
   MOV AH,09H
   INT 21H
   MOV AH,4CH
   INT 21H

COND2:
   LEA DX,S4
```

```
        MOV AH,09H
        INT 21H
        MOV AH,4CH
        INT 21H
    MAIN ENDP
END
```

edit: D:\STUDY\3rd year\3rd year 1st sem\System programming\a1\a1p07.ASM

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
01 ; Write and test a MASM program to validate second numbers is less than the first.
02
03 .MODEL SMALL
04 .STACK 100H
05 .DATA
06     S1 DB 0AH,0DH,"Enter 1st number: $"
07     S2 DB 0AH,0DH,"Enter 2nd number: $"
08     S3 DB 0AH,0DH,"2nd number is less than 1st number$"
09     S4 DB 0AH,0DH,"2nd number is greater than 1st number$"
10     S5 DB 0AH,0DH,"Two numbers are equal$"
11 .CODE
12     MAIN PROC
13         MOV AX,@DATA
14         MOV DS,AX
15
16         LEA DX,S1
17         MOV AH,09H
18         INT 21H
19
20         MOV AH,01H
21         INT 21H
22
23         MOV BL,AL
24         LEA DX,S2
25         MOV AH,09H
26         INT 21H
27
28         MOV AH,01H
29         INT 21H
30
31         CMP BL,AL
32         JG COND1
33         JL COND2
34         LEA DX,S5
35         MOV AH,09H
36         INT 21H
37
38         MOV AH,4CH
39         INT 21H
40
41         COND1:
42             LEA DX,S3
43             MOV AH,09H
44             INT 21H
45             MOV AH,4CH
46             INT 21H
47
48         COND2:
49             LEA DX,S4
50             MOV AH,09H
51             INT 21H
52             MOV AH,4CH
53             INT 21H
54         MAIN ENDP
55 END
```

emulator screen (80x25 chars)

```
Enter 1st number: 7
Enter 2nd number: 4
2nd number is less than 1st number
```

clear screen change font 0/16

46 INT 21H

Q8:

```
;Write and test a MASM program to find maximum and minimum from an array.
.model small
.stack 100h

.data
prompt_0  db  'enter the number of array elements :',0dh,0ah,'$'
prompt_1  db  'enter the array elements :',0dh,0ah,'$'
prompt_2  db  'the maximum is : $'
prompt_3  db  'the minimum is : $'

array   dw  50 dup(0)

s dw ?

.code
main proc

                mov ax, @data              ; initialize ds
                mov ds, ax

                lea dx, prompt_0           ; load and display the string prompt_0
```

```
mov ah, 9
int 21h

mov ah,1                    ;for taking input
int 21h

input1:
cmp al,0dh                  ;compare whether the pressed key is
'enter' or not
je line1                    ;if it is equal to 'enter' then stop taking
first value

and al,0fh                  ;convert it's ascii value to real value by
masking

shl bx, 1
shl bx, 1
shl bx, 1
shl bx, 1
or  bl,al                   ;making 'or' will add the current value
with previous value

int 21h
```

```
jmp input1

line1:
lea dx, prompt_1          ; load and display the string prompt_1
mov ah, 9
int 21h

lea si, array             ; set si=offset address of array

mov cx, bx                ; set cx=bx

@read_array:              ; loop label

mov ah,1                          ;for taking input
int 21h

xor dx,dx

input2:
cmp al,0dh                        ;compare whether the pressed key is
'enter' or not
```

```
        je line2                    ;if it is equal to 'enter' then stop taking
first value

        and al,0fh                  ;convert it's ascii value to real value by
masking

        shl dx,1
        shl dx,1
        shl dx,1
        shl dx,1
        or  dl,al                   ;making 'or' will add the current value
with previous value

        int 21h
        jmp input2

        line2:
        mov [si], dx          ; set [si]=ax
        add si, 2             ; set si=si+2

        mov dl, 0ah           ; line feed
        mov ah, 2             ; set output function
        int 21h               ; print a character
```

```
loop @read_array            ; jump to label @read_array while cx!=0
; array input done

lea si,array
mov ax,bx
dec ax
xor bx,bx
xor cx,cx
mov bx,word ptr[si]         ;store the maximum
mov cx,word ptr[si]         ;store the minimum
add si, 2

; loop to find max and min
arrayloop2:

cmp word ptr[si],bx
jg maximum

cmp word ptr[si],cx
jl minimum
```

```
jmp incre
maximum:
mov bx,word ptr[si]
jmp incre

minimum:
mov cx,word ptr[si]

incre:
add si, 2
dec ax

jnz arrayloop2

; displaying the prompt
lea dx,prompt_2
mov ah,09h
int 21h

; display contents of bx
output:                    ;level for printing their sum
```

```
mov dh,bh
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh

add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bh
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
shr dh, 1
shr dh, 1
```

```
shr dh, 1
shr dh, 1
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
and dh,0fh
cmp dh,10
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dl, 0ah              ; line feed
mov ah, 2                ; set output function
int 21h                  ; print a character

; displaying the prompt
lea dx,prompt_3
```

```
mov ah,09h
int 21h

; display contents of cx
mov bx,cx

mov dh,bh
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh

add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bh
and dh,0fh
add dh,'0'
mov dl,dh
```

```
mov ah,2
int 21h

mov dh,bl
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
and dh,0fh
cmp dh,10
add dh,'0'
mov dl,dh
mov ah,2
int 21h
```

```
exit:
mov ah, 4ch                    ;return control to dos
int 21h

main endp
end main
```

edit: D:\STUDY\3rd year\3rd year 1st sem\System programming\a1\a1p08.ASM

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
001 ;Write and test a MASM program to find maximum and minimum from an array.
002
003 .model small
004 .stack 100h
005
006 .data
007 prompt_0  db  'enter the number of array elements :',0dh,0ah,'$'
008 prompt_1  db  'enter the array elements :',0dh,0ah,'$'
009 prompt_2  db  'the maximum is : $'
010 prompt_3  db  'the minimum is : $'
011
012 array   dw  50 dup(0)
013
014 s dw ?
015
016 .code
017 main proc
018
019         mov ax, @data                   ; initialize ds
020         mov ds, ax
021
022         lea dx, prompt_0                ; load and display the string prompt
023         mov ah, 9
024         int 21h
025
026         mov ah,1
027         int 21h
028
029         input1:
030         cmp al,0dh
031         je line1
032
033         and al,0fh
034
035         shl bx, 1
036         shl bx, 1
037         shl bx, 1
038         shl bx, 1
039         or  bl,al
040
041         int 21h
042         jmp input1
043
044         line1:
045         lea dx, prompt_1
046         mov ah, 9
047         int 21h
048
049         lea si, array
050
051         mov cx, bx
052
053
054         @read_array:
055
056         mov ah,1
057         int 21h
```

emulator screen (80x25 chars)

```
enter the number of array elements:
enter the array elements:
5
4
7
2
the maximum is: 0007
                    the minimum is: 0002
```

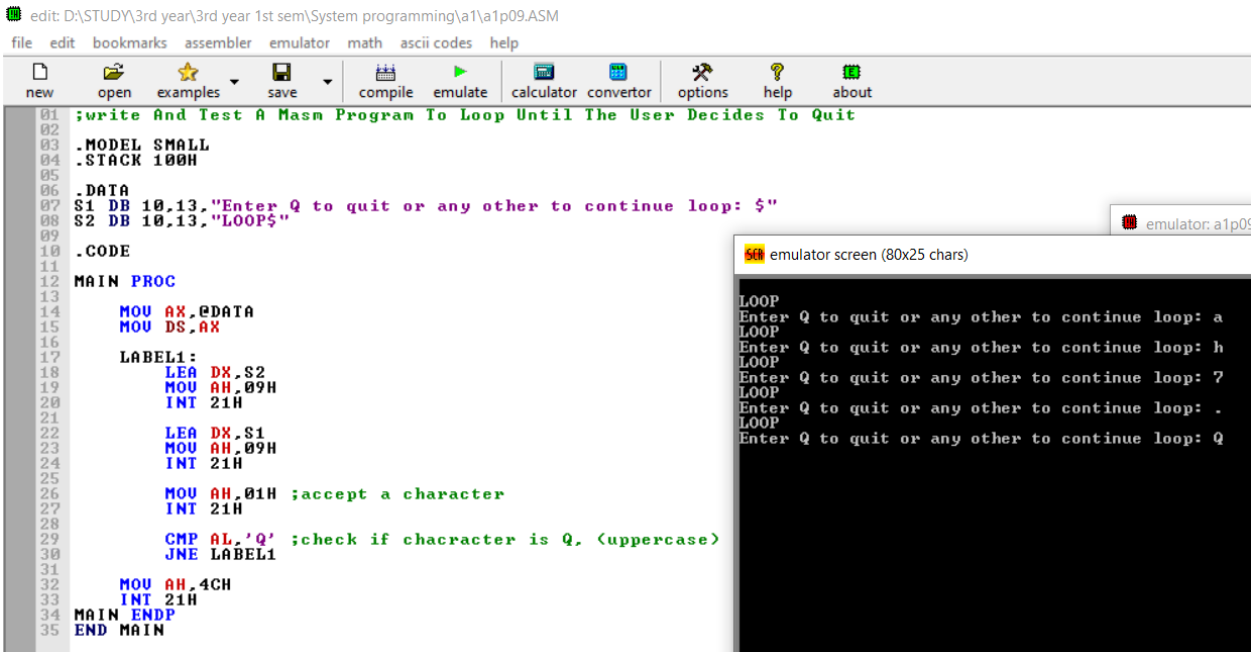
clear screen | change font | 0/16

;for taking input

Q9:

```
;Write and test a MASM program to loop until the user decides to quit

.MODEL SMALL
.STACK 100H

.DATA
S1 DB 10,13,"Enter Q to quit or any other to continue loop: $"
S2 DB 10,13,"LOOP$"

.CODE

MAIN PROC

    MOV AX,@DATA
    MOV DS,AX

    LABEL1:
        LEA DX,S2
        MOV AH,09H
        INT 21H
```

```
        LEA DX,S1
        MOV AH,09H
        INT 21H

        MOV AH,01H ;accept a character
        INT 21H

        CMP AL,'Q' ;check if chacracter is Q, (uppercase)
        JNE LABEL1

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



Q10:

; Write and test a MASM program to print all the characters from A-Z.

```
.MODEL SMALL

.STACK 100H

.DATA

   SPACE DB ‘ ‘

.CODE

MAIN PROC
```

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

MOV BX,65
MOV CX,0

LABEL1:
        MOV AH,02H  ;print A-Z
        MOV DL,BL
        INT 21H

        MOV AH,02H  ;print spaces
        MOV DL,SPACE
        INT 21H

        INC BX
        INC CX
        CMP CX,26   ;compare if space count is 26
JNE LABEL1

MOV AH,4CH
INT 21H
```

MAIN ENDP

END