8086 program pratice

```
.model small
.stack 100H
.data
A DB ?
B DB ?
MSG1 DB 'Enter number: $'
MSG2 DB 'Sum is: $'
SUM DB ?
.code
main proc
   mov Ax,@DATA
   mov Ds,AX

   mov AH,9
   lea DX,MSG1
   int 21h


   mov AH,1
   int 21h
   mov A,AL


   mov DL,0Dh
   mov AH,2
   int 21h
```

```
mov DL,0Ah

mov AH,2

int 21h



mov AH,9

lea DX,MSG1

int 21h


mov AH,1

int 21h

mov B,AL


mov DL,0Ah

mov AH,2

int 21h

mov DL,0Dh

mov AH,2

int 21h



mov AL,A

Add AL,B

mov SUM,AL
```

```
;converting SUM to decimal
;mov AX,20h
;sub AX,SUM
;mov SUM,AX


mov AH,9
lea DX,MSG2
int 21h



;converting asii of sum to ascii of number
mov AL,SUM
sub AL,30h
mov BL,AL
mov SUM,BL


mov AH,2
mov DL,SUM
int 21h




mov AH,4CH
int 21H
```

```
main ENDP
```

```
;prompt the user to enter a line of text. on the next line, display the
;capital letter entered that comes first alphabetically and the one
;that comes last. if no capital letters are entered, display
;"No Capital letters"

.model small
.stack 100H
.data
label1 DB "Enter line of text: $"
label2 DB "No Capital letters! $"
label3 DB "Capital letters are(Alphabetically): $"
label4 DB "Capital letters are: $"
text DB 11H DUP(?)
capital_letter DB 11H DUB(?)

.code
main proc
    mov AX,@DATA
    mov DS,AX
mov AH,9
lea DX,label1
int 21H
mov DL,0DH
mov AH,2
int 21H
mov DL, 0AH
mov AH,2
int 21H

mov AH,1
int 21H

mov DI,00H

WHILE:
   cmp AL,0DH
   je ENDWHILE

   mov text[DI],AL

   INC DI

   int 21H
   jmp WHILE

ENDWHILE:
```

```
mov text[DI],'$'

mov DL,0DH
mov AH,2
int 21H
mov DL, 0AH
mov AH,2
int 21H

mov DI,00H
mov AH,2
mov SI,00H

WHILE1:
    mov DL,text[DI]
    cmp DL,'$'
    je ENDWHILE1

    cmp DL,41H
    jl SKIP
    cmp DL,5AH
    jg SKIP

    mov capital_letter[SI],DL
    inc SI



    SKIP:
    inc DI
    ;int 21H

    jmp WHILE1

ENDWHILE1:

mov capital_letter[SI],'$'
cmp SI,0000H
je NOCAPITAL

mov DL,0DH
mov AH,2
int 21H
mov DL, 0AH
mov AH,2
int 21H

mov AH,9
lea DX,label4
int 21H
lea DX,capital_letter
int 21H



mov DI,00H
mov AH,2

WHILE2:
   mov BL,capital_letter[DI]
```

```
    cmp BL,'$'
    je ENDWHILE2
    inc DI
    mov SI,DI
    dec DI

    WHILE3:
        mov CL,capital_letter[SI]
        cmp CL,'$'
        je ENDWHILE3
        cmp BL,CL
        jl SKIP1
        mov capital_letter[DI],CL
        mov capital_letter[SI],BL
        mov BL,capital_letter[DI]

        SKIP1:
        inc SI
        jmp WHILE3
    ENDWHILE3:
    inc DI
    ;int 21H
    jmp WHILE2

ENDWHILE2:


mov DL,0DH
mov AH,2
```

```
int 21H
mov DL, 0AH
mov AH,2
int 21H

mov AH,9
lea DX,label3
int 21H

mov AH,9
;lea DX,text
lea DX,capital_letter;
int 21H

jmp EXIT

NOCAPITAL:
mov AH,9
;lea DX,text
lea DX,label2;
int 21H

EXIT:





mov AH,4CH
```

```
   int 21H

main ENDP



;STORE HEX CODE ENTERED INTO BX REGISTER WITH SHIFTING

.MODEL SMALL
.STACK 100H
.DATA
LABEL1 DB "Enter Hex Code: $"
.CODE
MAIN PROC
   MOV AX,@DATA
   MOV DS,AX
   MOV BX,0000H
   MOV CL,4

   MOV AH,9
   LEA DX,LABEL1
   INT 21H

   MOV AH,1
   INT 21H
   WHILE:
      CMP AL,0DH
      JE ENDWHILE
      CMP AL,39H
      JG LETTER

      AND AL,0FH
      JMP SHIFT

      LETTER:
         SUB AL,37H

      SHIFT:
         SHL BX,CL
         OR BL,AL

      INT 21H
      JMP WHILE

   ENDWHILE:

   MOV AH,4CH
   INT 21H
MAIN ENDP




;generate mutiplication table

.model small
```

```
.stack 100h
.data
NUM DB ?
LABEL1 DB "Enter number: $"

.code
main proc

    mov AX,@DATA
    mov DS,AX

    mov AH,9
    lea DX,LABEL1
    int 21H

    mov AH,1
    int 21H
    mov BL,AL

    mov BH,BL
    sub BH,30h ;convert ascii to num



    mov CL,00h

    TABLE:
        inc CL
        cmp CL,10d
        je DONE

        mov DL,0DH
        mov AH,2
        int 21H
        mov DL, 0AH
        int 21H
        mov DL,BL
        int 21h
        mov DL,'x'
        int 21h
        mov DL,CL
        add DL,30h
        int 21H
        mov DL,'='
        int 21h

        mov AL,CL
        mul BH

        call DISPLAY_DIGIT



        jmp TABLE

    DONE:

    ;mov Al,0fh
```

```
    ;mov bl,10h
    ;mul bl

    ;mov DI,AX




    mov AH,4CH
    int 21h

main ENDP


DISPLAY_DIGIT proc

    ;initilize count
    mov DI,0
    mov DX,0
    label3:
        ; if ax is zero
        cmp AX,0
        je print1

        ;initilize bx to 10
        mov SI,10

        ; extract the last digit
        div SI
```

```
   ;push it in the stack
   push DX

   ;increment the count
   inc DI

   ;set dx to 0
   xor DX,DX
   jmp label3
print1:
   ;check if count
   ;is greater than zero
   cmp DI,0
   je exit

   ;pop the top of stack
   pop DX

   ;add 48 so that it
   ;represents the ASCII
   ;value of digits
   add DX,48

   ;interuppt to print a
   ;character
   mov AH,02h
   int 21h

   ;decrease the count
```

```
        dec DI
        jmp print1

    exit:

        RET
DISPLAY_DIGIT ENDP

END main
```

```
; this program alternate the case of letters

.model small
.stack 100H
.data
label1 DB "Enter line of text: $"
label3 DB "Line of text after case conversion: $"
text DB 11H DUP(?)
inverted_letters DB 11H DUB(?)

.code
main proc
    mov AX,@DATA
    mov DS,AX
mov AH,9

lea DX,label1

int 21H

mov DL,0DH

mov AH,2

int 21H

mov DL, 0AH

mov AH,2

int 21H


mov AH,1

int 21H


mov DI,00H


WHILE:

   cmp AL,0DH

   je ENDWHILE


   mov text[DI],AL


   INC DI


   int 21H

   jmp WHILE


ENDWHILE:


mov text[DI],'$'
mov DL,0DH
mov AH,2
int 21H
mov DL, 0AH
mov AH,2
int 21H

mov DI,00H
mov AH,2
mov SI,00H

WHILE1:
    mov DL,text[DI]
    cmp DL,'$'
    je ENDWHILE1

    cmp DL,41H
    jl SKIP
    cmp DL,5AH
    jle TOLOWER

    cmp DL,61H
    jl SKIP
    cmp DL,7AH
    jle TOUPPER

    jmp SKIP
    TOUPPER:
      sub DL,20h
      mov inverted_letters[SI],DL
      inc SI
      jmp DONE

    TOLOWER:
      add DL,20h
      mov inverted_letters[SI],DL
      inc SI
      jmp DONE



    SKIP:
      mov inverted_letters[SI],DL
      inc SI

    DONE:
       inc DI
       jmp WHILE1

ENDWHILE1:

mov inverted_letters[SI],'$'

mov DL,0DH
mov AH,2
int 21H
```

```
    mov DL, 0AH
    mov AH,2
    int 21H

    mov AH,9
    lea DX,label3
    int 21H
    lea DX,inverted_letters
    int 21H


    mov DI,00H
    mov AH,2


    EXIT:




    mov AH,4CH
    int 21H

main ENDP
```