1. **Write an assembly language program to add two numbers.**

```
TITLE ADDITION OF TWO NUMBERS
.MODEL SMALL
.STACK 64
.DATA
NUM1 DB 12
NUM2 DB 24
SUM DB ?
.CODE
MAIN PROC FAR
MOV AX,@data
MOV DS,AX

MOV AL,NUM1
ADD AL,NUM2
MOV SUM,AL

MOV AX,4C00H
INT 21H

MAIN ENDP
END MAIN
```

2. **Write an assembly language program to sum numbers from 0 to 255.**

```
TITILE Write an assembly language program to sum numbers from 0 to 255
.MODEL SMALL
.STACK 64
.DATA
SUM DW 0
.CODE
MAIN PROC FAR
   MOV AX,@DATA
   MOV DS,AX

   MOV CX,255
   ADDITION: ADD SUM,CX
   LOOP ADDITION

   MOV AX,4C00H
   INT 21H

   MAIN ENDP
END MAIN
```

3. **Write an assembly language program to add ten 16 bit numbers stored in memory and store the result.**

```
TITLE ADDITION OF TEN 16 BIT NUMBERS
.MODEL SMALL
.STACK 64
.DATA
ARRNUM DW 125,2435,123,6789,456,12,14,123,123,124
SUM DW 0

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

   MOV CX,10
   MOV SI,OFFSET ARRNUM
  NEXTDATA: MOV BX,[SI]
   ADD SUM,BX
   INC SI
   LOOP NEXTDATA


   MOV AX,4C00H
   INT 21H

   MAIN ENDP
END MAIN
```

4. **There are two tables each consist ten 16 bit data at memory. Write a program to add data from two tables of respective index value and store in third table.**

```
TITLE ADDITION OF TEN 16 BIT NUMBERS OF TWO TABLES

.MODEL SMALL

.STACK 64

.DATA

ARR1 DW 125,2435,123,6789,456,12,14,123,123,124

ARR2 DW 345,342,789,123,456,45,21,34,45,678

ARRSUM DW 10 DUP(?)
```

```
.CODE
MAIN PROC FAR
   MOV AX,@DATA
   MOV DS,AX
    MOV CX,10
   MOV SI,OFFSET ARR1
   LEA DI,ARR2
   LEA BX,ARRSUM

  NEXTDATA: MOV AX,[SI]
   ADD AX,[DI]
   MOV [BX],AX

   INC SI
   INC DI
   INC BX
   LOOP NEXTDATA
   MOV AX,4C00H
   INT 21H
   MAIN ENDP
END MAIN
```

5. **Write a program to display a string "I Love Programming" in the screen using string display function.**

```
.model small
.stack 64
.data
```

```
str db "I Love Programming$"

.code
main proc far
   mov ax,@data
   mov ds,ax

   lea dx,str
   mov ah,09h
   int 21h

   mov ax,4c00h
   int 21h
   main endp
end main
```

6. **Write a program to display a string "I Love Programming" in the screen using character display function.**

```
.model small
.stack 64
.data
str db "I Love Programming $"

.code
main proc far
   mov ax,@data
   mov ds,ax
   lea si,str
   mov ah,02h
   display:
   mov dl,[si]
   cmp dl,'$'
   je down
   int 21h
   inc si
   jmp display
  down: mov ax,4c00h
   int 21h
   main endp
end main
```

7. Write a program to read string from the user and display the string in another line by converting into uppercase if it is in lowercase.

```
.model small
.stack 64

.data
str1 db "enter your string",0dh,0ah,"$"
line db 0dh,0ah,"$"
max db 100
act db ?
str2 db 100 dup("$")

.code
main proc far
   mov ax,@data
   mov ds,ax

   lea dx,str1
   mov ah,09h
   int 21h

   lea dx,max
   mov ah,0ah
   int 21h

   lea dx,line
   mov ah,09h
   int 21h

   lea si,str2
   mov ah,02h
   mov ch,00
   mov cl,act
   l1:mov dl,[si]
   cmp dl,97
   jb down
   cmp dl,122
   ja down
   sub dl,32
   down:int 21h
```

```
        inc si
        loop l1
    mov ax,4c00h
    int 21h
    main endp
end main
```

8. **Write a program that takes a string from user and count the number of vowels in string and display it in decimal format.**

```
.model small
.stack 64h
.data
        str1 db "enter any string$"
        str2 db "total no of vowel = $"
        count db 0
        line db 0ah,0dh,'$'
        max db 100
        act db ?
        str3 db 100 dup('$')
        vowel db "aeiouAEIOU"

.code
        main proc far
        mov ax,@data
        mov ds,ax

        lea dx,str1
        mov ah,09h
        int 21h

        call nextline

        lea dx,max
        mov ah,0ah
        int 21h

        call nextline

        mov cl,act
        mov ch,0
```

```
mov bx,0

l1:push cx
mov al,str3[bx]
mov si,0
mov cx,10

l2:cmp al,vowel[si]
jz l3
inc si
loop l2
jmp l4

l3:inc count
l4:inc bx
pop cx
loop l1

lea dx,str2
mov ah,09h
int 21h

mov cx,10
mov al,count
mov ah,0
mov bx,0
l5:mov dx,0
div cx
add dx,30h
push dx
inc bx     ;count no. of digit
cmp ax,0
ja l5

mov ah,02
mov cx,bx
disp:pop dx
int 21h
loop disp
```

```
mov ax,4c00h
int 21h
main endp

nextline PROC NEAR
   lea dx,line
   mov ah,09h
   int 21h
   ret
   nextline endp

end main
```

9. Write a program to find the sum of numbers from 1 to n. read n from user and display the sum in decimal format.

```
.model small
.stack 64h
.data
        str1 db "enter value of n",0ah,0dh,"$"
        str2 db "sum from 1 to n = ","$"
        num db 0
        count db 0
        sum dw 0000h
        line db 0ah,0dh,'$'

.code
        main proc far
        mov ax,@data
        mov ds,ax

        lea dx,str1
        mov ah,09h
        int 21h


        lea si,num

        l1:mov ah,01h
        int 21h
```

```
cmp al,0dh
je l2
sub al,30h
push ax
mov al,10
mul num
mov num,al
pop ax
add num,al
inc count
jmp l1

l2:mov dl,num
mov dh,0
l3:add sum,dx
dec dl
jnz l3

lea dx,line
mov ah,09h
int 21h

lea dx,str2
mov ah,09h
int 21h

mov cx,10
mov ax,sum
mov bh,0
l4:mov dx,0
div cx
add dx,30h
push dx
inc bx
cmp ax,0
ja l4

mov ah,02
mov cx,bx
disp:pop dx
```

```
        int 21h
        loop disp

        mov ax,4c00h
        int 21h
        main endp
        end main
```

10. Write a program to read a number from user and display the multiplication table of a number on screen.

```
.model small
.stack 64
.data
str1 db "Enter number$"
str2 db " x $"
str3 db " = $"
str4 db "***Multiplication Table***$"
line db 0dh, 0ah, '$'
num db ?


.code
main proc far
   mov ax,@data
   mov ds,ax

   lea dx,str1
        mov ah,09h
        int 21h

        call newline

        call input_num

   call newline

   lea dx,str4
   mov ah,09h
   int 21h
```

```
    call newline

    mov bx,01
    mov cx,10

    l3:mov ax,bx
       mul num
       call disp_table
       call disp_num
       call newline
       inc bx
       loop l3

    mov ax,4c00h
    int 21h
    main endp

input_num proc near

          mov ah,01h
          int 21h

          sub al,30h
          mov num,al
          ret
          input_num endp
disp_num proc near

    push bx
    push cx
          mov bx,0
          l5:mov cx,10
          mov dx,0
          div cx
          add dx,30h
          push dx
          inc bx
          cmp ax,0
          ja l5
```

```
        mov ah,02
        mov cx,bx
        disp:pop dx
        int 21h
        loop disp
        pop cx
        pop bx
        ret
        disp_num endp

newline proc near
   push dx
   push ax
   lea dx,line
   mov ah,09h
   int 21h
   pop ax
   pop dx
   ret
   newline endp

disp_table proc near
   push dx
   push ax
   mov ah,0
   mov al,num
   call disp_num

    lea dx,str2
    mov ah,09h
    int 21h

    mov ax,bx
    call disp_num

    lea dx,str3
    mov ah,09h
    int 21h
    pop ax
   pop dx
```

```
    ret
    disp_table endp

end main
```

11. Write a program to read string from user and display that string in a clear screen with reverse order.

```
.model small
.stack 64
.data
str1 db "Enter a string$"
max db 100
act db ?
str2 db 100 dup('$')
line db 0dh,0ah,'$'
.code
main proc far
   mov ax,@data
   mov ds,ax
   ;disply str1
   lea dx,str1
   mov ah,09h
   int 21h
   ;newline
   lea dx,line
   mov ah,09h
   int 21h
   ;read string from user
   lea dx,max
   mov ah,0ah
   int 21h
   ;clear screen
   mov ax,0000h
   int 10h
   ;disply characterwise in reverse
   mov cl,act
   mov ch,0
   lea si,str2
```

```
    add si,cx      ;set si to end of string

    up:dec si
    mov dl,[si]
    mov ah,02h     ;disply character
    int 21h
    loop up
    ;terminate program
    mov ax,4c00h
    int 21h
    main endp
end main
```

12. Write a program to print the word "Hello" in the center of the screen with green color text in blue background.

```
.model small
.stack 64h
.data
str db "Hello$"

.code
main proc far
    mov ax,@data
    mov ds,ax

    ;clear the screen
    mov ah,6   ;scroll up
    mov al,0   ;full screen
    mov cl,0 ;start point of column
    mov ch,0 ;start point of rows
    mov dl,79 ;end point of column
    mov dh,24 ;end point of rows
    mov bh,12h  ;black BG green FG
    int 10h

    ;position the curser in center
    mov ah,02h ;set cursor position
    mov bh,0    ;page number 0
```

```
    mov dl,39 ;mid point of columns
    mov dh,12 ;mid point of rows
    int 10h

    ;print the message
    lea dx,str
    mov ah,09h
    int 21h
    ;end of program
    mov ah,4ch
    int 21h
    main endp
end main
```

13. Write a program to clean the lower part of the screen with a green background and print "Hello" in the middle of it with a light red background and white foreground.

```
.model small
.stack 64h
.data
str db "Hello$"
num db 5
.code
main proc far
    mov ax,@data
    mov ds,ax
    ;clear screen
    mov ah,6  ;scroll up
    mov al,0   ;full screen
    mov cl,0 ;start point of column
    mov ch,0ch ;start point of rows
    mov dl,4fh ;end point of columns
    mov dh,18h ;end point of rows
    mov bh,20h ;Green BG black FG
    int 10h

    mov ah,6
    mov al,0
    mov cl,27h ;start of the colums
    add num,cl
    dec num
```

```
    mov ch,0ch ;start point of rows
    mov dl,num ;end point of columns
    mov dh,0ch ;end point of rows
    mov bh,0cfh ;light red BG white FG
    int 10h

    mov ah,2
    mov bh,0
    mov dl,27h ;midpoint of columns
    mov dh,0ch ;midpoint of rows
    int 10h

    lea dx,str
    mov ah,09h
    int 21h

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