8086 ALPs

1. Write an assembly language program for 8086 to read a character from user and display it.
2. Write an assembly language program for 8086 to read a string and display on the screen.
3. Write an assembly language program for 8086 to read string and print it in the reverse order.
4. Write an assembly language program for 8086 to read a string and display the string in uppercase.
5. Write an assembly language program for 8086 to read a string and display the string in lowercase.
6. Write an assembly language program for 8086 to print number of vowels in a given string.
7. Write an assembly language program for 8086 to read two strings and check whether they are same or not.
8. Write an assembly language program for 8086 to concatenate two strings.
9. Write an assembly language program for 8086 input String and Display Each Word in Next Line

Solutions:

1)

```
include 'emu8086.inc'
.stack 100h
.model small

.data
    char db ?
.code
    main proc
        mov ax, @data
        mov ds, ax

        print 'Enter a character: '

        mov ah, 01h
        int 21h

        mov char, al

        mov dl, 10      ;for newline
        mov ah, 02h
        int 21h

        mov dl, 13      ;for carriage return
        mov ah, 02h
        int 21h

        print 'Your character is: '

        mov dl, char
        mov ah, 02h
        int 21h
    main endp
end main
```

2)

```
.model small
.stack
.data
    name1 db 13, 10, 'Tribhuvan University$'
    name2 db 13, 10, 'Kathmandu Nepal$'

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

        lea dx, name1
        mov ah,09h
        int 21h

        lea dx, name2
        mov ah,09h
        int 21h
    main endp
end main
```

3)

```
include 'emu8086.inc'
.stack 100h
.model small
.data
    string db 'TEXAS CSIT SECOND SEMESTER$'
.code

    main proc
        mov ax, @data
        mov ds, ax

        print 'Original String: '
        lea dx, string
        mov ah, 09h
        int 21h

        mov dl, 10
        mov ah, 02h
        int 21h

        mov dl, 13
        mov ah, 02h
        int 21h

        mov si, offset string
        mov cx,26

        print 'string after reversal: '
        l1:
            mov bx, [si]
            push bx
            inc si
            loop l1

         mov cx, 26
         l2:
            pop dx
            mov ah, 02h
            int 21h
            loop l2

    main endp
end main
```

4)

```
.model small
.stack
.data
    msg1 db 10, 13, 'Original String is:$'
    msg2 db 10, 13, 'Uppercase String is:$'
    name1 db 'tribhuvan university$'
.code
    main proc
        mov ax, @data
        mov ds, ax
        ;.startup

        lea dx, msg1
        mov ah, 09h
        int 21h

        mov dx,offset name1
        mov ah,09h
        int 21h

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

        mov cx,19
        mov si, offset name1

        uppercase:
        cmp [si], 32
        jne skip

        mov dl, [si]
        mov ah, 02h
        int 21h
        inc si

        skip:
        sub [si],20h
        mov dl,[si]
        mov ah,02h
        int 21h
        inc si

        loop uppercase

        ;.exit
    main endp
end main
```

5)

```
.model small
.stack
.data
    msg1 db 10, 13, 'Original Text is:$'
    msg2 db 10, 13, 'Lowercase Text is:$'
    string db 'TRIBHUVAN UNIVERSITY$'
.code
    main proc
        mov ax, data
        mov ds, ax

        lea dx, msg1
        mov ah, 09h
        int 21h

        mov dx, offset string
        mov ah,09h
        int 21h

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

        mov cx,19
        mov si, offset string

        lowercase:
        cmp [si], 32
        jne skip
        mov dl, [si]
        mov ah, 02h
        int 21h
        inc si
        skip:
        add [si],20h
        mov dl,[si]
        mov ah,02h
        int 21h
        inc si

        loop lowercase
    main endp
end main
```

6)

```
.model small
.stack
.data
    msg db 10, 13, 'Number of vowels in the given string is: $'
    string db 'mIcroprocEsOr$'
.code
    main proc
        mov ax, @data
        mov ds, ax
        ;.startup
        mov bh,0        ;to count no. of vowels
        mov cx,14       ;length of string
        mov si,offset string

        AGAIN:
        mov al,[si]
        cmp al, 61h  ;ASCII of ‘a’ in HEX
        je COUNT ; jump if equal

        cmp al, 41h  ;ASCII of ‘A’ in HEX
        je COUNT ; jump if equal

        cmp al, 65h  ;ASCII of ‘e’ in HEX
        je COUNT ; jump if equal

        cmp al, 45h  ;ASCII of ‘E’ in HEX
        je COUNT ; jump if equal

        cmp al, 69h  ;ASCII of ‘I’in HEX
        je COUNT ; jump if equal

        cmp al, 49h  ;ASCII of ‘I’ in HEX
        je COUNT ; jump if equal

        cmp al, 6fh  ;ASCII of ‘o’ in HEX
        je COUNT ; jump if equal

        cmp al, 4fh  ;ASCII of ‘O’ in HEX
        je COUNT ; jump if equal
        cmp al, 75h  ;ASCII of ‘u’ in HEX
        jne SKIP ; jump if not equal

        cmp al, 49h  ;ASCII of ‘U’ in HEX
        je COUNT ; jump if equal

        COUNT: inc bl

        SKIP: inc si
        loop AGAIN

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

        add bl,30h       ;convert count to ASCII to print
        mov dl,bl
        mov ah,02h
        int 21h

        main endp
      end main
```

7)

```
    .model small
    .stack 100h
    .data
      cr equu 13
      nl equ 10

      inmsg1  db cr,nl, 'enter the string1: $'
      inmsg2  db cr,nl,nl, 'enter the string2: $'

      strng1  db 0bh,12 dup(?)
      strng2  db 0bh,12 dup(?)

      sucmsg  db cr,nl,nl,'both are same $'
      falmsg  db cr,nl,nl,'different strings $'

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

           lea dx,inmsg1
           mov ah,09
           int 21h

           mov dx,offset strng1
           mov ah,0ah
```

```
            int 21h

            lea dx,inmsg2
            mov ah,09
            int 21h

            mov dx,offset strng2
            mov ah,0ah
            int 21h

            mov si, offset strng1
            mov di, offset strng2
            cld

            mov cx,6h
            repe cmpsb
            jz success

            lea dx,falmsg
            jmp display

            success: lea dx,sucmsg

            display: mov ah,09
            int 21h

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

8)

```
 .model small
.stack
.data
    string1 db 'Microprocessor is an$'
    string2 db 'Assembly Language.$'
    string3 db ?
    spc equ 32

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

        mov di,offset string3
        mov si,offset string1

        mov cx,20

        l1:
        mov bx,[si]
    mov [di],bx
    inc si
    inc di
    loop l1

    mov [di],spc    ;To print space between two strings
    inc si
    inc di

    mov si,offset string2
    mov cx,19
    l2:
    mov bx,[si]
    mov [di],bx
    inc si
    inc di
    loop l2

    mov dx, offset string3
    mov ah,09h
    int 21h

    main endp
end
```

9)

```
include 'emu8086.inc'
.model small
.stack 100
.data
  msg db 60 dup(?)

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

  lea si, msg
  print 'Enter your string: '

  input:
    mov ah,1
    int 21h
    cmp al,13
    je display
    mov [si],al
    inc si
    jmp input

  display:
    mov [si],'$'
    lea di, msg
```

```
        mov dl,10
        mov ah,2
        int 21h
        mov dl,13
        mov ah,2
        int 21h

    again:
        cmp [di],'$'
        je last
        cmp [di],32
        je next
        mov dl,[di]
        mov ah,2
        int 21h
        inc di
        jmp again

    next:
        mov dl,10
        mov ah,2
        int 21h

        mov dl,13
        mov ah,2
        int 21h

        inc di
        jmp again

    last:
    mov ah,4ch
    int 21h

    main endp
end
```

Assignments

1. Write an assembly language program for 8086 to calculate factorial of a given number.
2. Write an assembly language program for 8086 to find the sum of Natural numbers from 1 to 10. [1+2+3+….+10]
3. Write an assembly language program for 8086 to print the multiplication table of a given number.