```
%macro scall 4 ;macro declaration with 4 parameters

    mov eax,%1  ;1st parameter has been moved to eax
    mov ebx,%2  ;2nd parameter has been moved to ebx
    mov ecx,%3  ;3rd parameter has been moved to ecx
    mov edx,%4  ;4th parameter has been moved to edx
    int 80h    ;Call the Kernel
%endmacro       ;end of macro

section .data  ;.data begins here
    m1 db 10d,13d,"Enter a string: "  ;m1 variable initialized with string
    l1 equ $-m1                       ;l1 stores length of string m1

    m2 db 10d,13d,"Entered String: "   ;m2 variable initialized with string
    l2 equ $-m2                       ;l2 stores length of string m2

    m3 db 10d,13d,"Length: "           ;m3 variable initialized with string
    l3 equ $-m3                       ;l3 stores length of string m3

section .bss   ;.bss begins here
    buffer resb 50         ;buffer array of size 50
    size equ $-buffer      ;size variable to have input
    count resd 1           ;to store size of buffer
    dispnum resb 8         ;to display 8 digit length

section .text  ;.text begins here
    global _start         ;moving to _start label

_start:                    ;_start label
    scall 4,1,m1,l1        ;macro call to display m1
    scall 3,0,buffer,size  ;macro call to input buffer
    mov [count],eax        ;length of buffer gets stored in count
    scall 4,1,m2,l2        ;macro call to display m2
    scall 4,1,buffer,[count] ;macro call to display buffer
    scall 4,1,m3,l3        ;macro call to display m3

    mov esi,dispnum+7      ;esi points to 8th location of dispnum
    mov eax,[count]        ;eax now stores value of count
    mov ecx,8             ;ecx gets initialized with 8
    dec eax                ;decrement the value of eax

UP1:                        ;UP1 label
    mov edx,0             ;edx gets initialized with 0
    mov ebx,10            ;ebx gets initialized with 10
    div ebx               ;divide the contents of eax by ebx
    add dl,30h            ;add 30 to the remainder
    mov [esi],dl          ;dl content gets copied at esi
    dec esi               ;decrement esi
    loop UP1              ;jump to UP1 till ecx becomes 0

```

```
50      scall 4,1,dispnum,8    ;macro call to display dispnum array
51
52      mov eax,1              ;sys_exit function
53      mov ebx,0              ;Successful Termination
54      int 80h                ;Call the Kernel
55
```

Output: