

1	%macro scall 4	;macro declaration with 4 parameters
2	mov eax,%1	;1st parameter has been moved to
3	eax	
4	mov ebx,%2	;2nd parameter has been moved to
5	ebx	
6	mov ecx,%3	;3rd parameter has been moved to
7	ecx	
8	mov edx,%4	;4th parameter has been moved to
9	edx	
10	int 80h	;Call the Kernal
11	%endmacro	;end of macro
12	section .data	;.data begins here
13	m1 db 10d,13d,"Enter a string: "	;m1 variable initialised with string
14	l1 equ \$-m1	;l1 stores length of string m1
15	m2 db 10d,13d,"Entered String: "	;m2 variable initialised with string
16	l2 equ \$-m2	;l2 stores length of string m2
17	m3 db 10d,13d,"Length: "	;m3 variable initialised with string
18	l3 equ \$-m3	;l3 stores length of string m3
19	section .bss	;.bss begins here
20	buffer resb 50	;buffer array of size 50
21	size equ \$-buffer	;size variable to have input
22	count resd 1	;to store size of buffer
23	dispnum resb 8	;to display 8 digit length
24	section .text	;.text begins here
25	global _start	;moving to _start label
26	_start:	;_start label
27	scall 4,1,m1,l1	;macro call to display m1
28	scall 3,0,buffer,size	;macro call to input buffer
29	mov [count],eax	;length of buffer gets stored in count
30	scall 4,1,m2,l2	;macro call to display m2
31	scall 4,1,buffer,[count]	;macro call to display buffer
32	scall 4,1,m3,l3	;macro call to display m3
33	mov esi,dispnum+7	;esi points to 8th location of dispnum
34	mov eax,[count]	;eax now stores value of count
35	mov ecx,8	;ecx gets initiaised with 8
36	dec eax	;decrement the value of eax
37	UP1:	;UP1 label
38	mov edx,0	;edx gets initiaised with 0
39	mov ebx,10	;ebx gets initialised with 10
40	div ebx	;divide the contents of eax by ebx
41	add dl,30h	;add 30 to the remainder
42	mov [esi],dl	;dl content gets copied at esi
43	dec esi	;decrement esi
44	loop UP1	;jump to UP1 till ecx becomes 0
45	scall 4,1,dispnum,8	;macro call to display dispnum array
46	mov eax,1	;sys_exit function
47	mov ebx,0	;Sucessful Termination
48	int 80h	;Call the Kernal
49		
50		