**COA ASSIGNMENT NO. :- 4**

**Aim:-** Write 64-bit ALP to accept number and display it on screen.

**Objective:-** Here we have learnt how to accept a 64 bit number and display it.

**Theory:-**

An assembly program can be divided into three sections −

* The **data** section,
* The **bss** section, and
* The **text** section.

**The *data* Section**

The **data** section is used for declaring initialized data or constants. This data does not change at runtime. You can declare various constant values, file names, or buffer size, etc., in this section.

The syntax for declaring data section is −

section.data

**The *bss* Section**

The **bss** section is used for declaring variables. The syntax for declaring bss section is −

section.bss

**The *text* section**

The **text** section is used for keeping the actual code. This section must begin with the declaration **global \_start**, which tells the kernel where the program execution begins.

The syntax for declaring text section is −

section.text

global \_start

\_start:

**Source Code:-**

%macro scall 4

mov rax,%1

mov rdi,%2

mov rsi,%3

mov rdx,%4

syscall

%endmacro

section .data

m1 db "Enter 64 bit number is = $",10d,13d

l1 equ $-m1

m2 db " The 64 bit number is = $ ",10d,13d

l2 equ $-m2

section .bss

num resq 1

section .text

global \_start

\_start:

scall 1,1,m1,l1

scall 0,0,num,17

scall 1,1,m2,l2

scall 1,1,num,17

mov rax,60

mov rdi,0

syscall

**Output:-**

Enter 64 bit number is = 1234567812345678

The 64 bit number is = 1234567812345678