Assignment No.

Program:

```
;Assignment
;Title : Write X86/64 ALP to accept a string and to display its
length.
;Student Name - Aviraj Popat Kale
;PRN number : 122B1B123
;Perform Date :
;-------------------------section data----------------------
section    .data
    msg     db    10,10,"Enter the string:   "; Define a message to
prompt the user to enter a string
    msg_len equ   $-msg; Calculate the length of the message

    smsg    db    10,10,"The length of string is:   "; Define a
message to display the length of the string
    smsg_len equ  $-smsg; Calculate the length of the message

;-----------------------------bss section---------------------------
Section   .bss
    string  resb  50; Allocate space in memory to store the input
string (50 bytes)
    stringl equ   $-string; Calculate the length of the string

    count   resb  1; Allocate space in memory to store the count of
characters in the string
    char_ans resb  2; Allocate space in memory to store the ASCII
representation of the count

;---------------------------------------------------------------------
%macro  Print   2
    mov   rax, 1; Load the system call number for write (1) into rax
    mov   rdi, 1; Load file descriptor for stdout (1) into rdi
    mov   rsi, %1;Load the address of the message to be printed into
rsi
    mov   rdx, %2; Load the length of the message into rdx
    syscall; Invoke the kernel to perform the write operation
%endmacro

%macro  Read   2
    mov   rax, 0; Load the system call number for read (0) into rax
    mov   rdi, 0; Load file descriptor for stdin (0) into rdi
    mov   rsi, %1; Load the buffer address to store input into rsi
    mov   rdx, %2; Load the maximum number of bytes to read into rdx
    syscall; Invoke the kernel to perform the read operation
%endmacro

%macro  Exit    0
    mov  rax, 60; Load the system call number for exit (60) into rax
    mov  rdi, 0; Load the exit status (0) into rdi
    syscall; Invoke the kernel to terminate the program
%endmacro

;---------------------------code section---------------------------------
section    .text
    global   _start
_start:
    Print  msg, msg_len; Print the message prompting the user to enter a string
    Read   string, stringl; Read the input string from the user
    mov    [count], rax; Store the length of the input string

    Print  smsg, smsg_len; Print the message indicating the length of the string
    mov    rax, [count]; Load the length of the string into rax
    call   Display; Call the Display subroutine to display the length in ASCII
    Exit; Terminate the program
;---------------------------------------------------------------------
Display:
    mov    rbx,16; Set divisor to 16 for hexadecimal conversion
    mov    rcx,2; Set number of digits to 2 for displaying byte value

    mov    rsi,char_ans+1; Load the address of the last byte in char_ans buffer into rsi
cnt:
    mov    rdx,0; Clear rdx (as in div instruction rdx:rax/rbx)
    div    rbx; Divide rax by rbx (divisor)

    cmp    dl, 09h; Check if remainder (dl) is less than or equal to 9
    jbe    add30; Jump if less than or equal to 9
    add    dl, 07h; Adjust remainder for hexadecimal conversion

add30:
    add    dl,30h; Convert remainder to ASCII character
    mov    [rsi],dl; Store ASCII character in buffer
    dec    rsi; Move pointer to previous byte in buffer

    dec    rcx; Decrement digit count
    jnz    cnt; Repeat until all digits are processed

    Print  char_ans, 2; Print the ASCII representation of the count
    ret

;-----------------------------------------------------------------
```

Output:

```
Enter the string:Aviraj

The length of string is:   06
```