```
;syeda sharmeen asim shah

;assembly language program project on
;vowel quantitative tabulation program

section .data
    prompt db "Enter a string: ", 0
    result_msg db "Total vowels found: ", 0
    newline db 0Ah, 0

section .bss
    input_str resb 100    ; Reserve space for input string
    vowel_count resb 2    ; Reserve space for vowel count (to hold 2 ASCII
characters)

section .text
    global _start

_start:
    ; Display prompt
    mov eax, 4
    mov ebx, 1
    mov ecx, prompt
    mov edx, 16
    int 0x80

    ; Read user input into input_str
    mov eax, 3
    mov ebx, 0
    mov ecx, input_str
    mov edx, 100
    int 0x80

    ; Call function to count vowels
    xor esi, esi            ; Clear counter (ESI)
    xor bl, bl              ; Clear vowel_count register (BL)

count_vowels_loop:
    mov al, byte [ecx+esi]  ; Load a character from input_str
    cmp al, 0               ; Check if end of string (null character)
    je display_result

    cmp al, 'a'
    je increment_vowel_count
    cmp al, 'e'
    je increment_vowel_count
    cmp al, 'i'
    je increment_vowel_count
    cmp al, 'o'
    je increment_vowel_count
    cmp al, 'u'
    je increment_vowel_count
    cmp al, 'A'
    je increment_vowel_count
    cmp al, 'E'
    je increment_vowel_count
    cmp al, 'I'
    je increment_vowel_count
    cmp al, 'O'
```

```
    je increment_vowel_count
    cmp al, 'U'
    je increment_vowel_count

increment_index:
    inc esi                 ; Increment counter
    jmp count_vowels_loop

increment_vowel_count:
    inc bl                 ; Increment vowel_count register (BL)
    jmp increment_index

display_result:
    ; Convert the vowel count to ASCII characters
    add bl, '0'             ; Convert the count to ASCII
    mov [vowel_count], bl  ; Store the least significant digit

    ; Display result message
    mov eax, 4
    mov ebx, 1
    mov ecx, result_msg
    mov edx, 19
    int 0x80

    ; Display vowel_count
    mov eax, 4
    mov ebx, 1
    mov ecx, vowel_count
    mov edx, 2              ; Display 2 ASCII characters
    int 0x80

    ; Display newline character
    mov eax, 4
    mov ebx, 1
    mov ecx, newline
    mov edx, 2
    int 0x80

    ; Exit program
    mov eax, 1
    xor ebx, ebx
    int 0x80
```