PROBLEM STATEMENT:

Write an X86/64 ALP to accept five 64-bit hexadecimal numbers from user, store them in an array and display the accepted numbers.

SOURCE CODE:

```
section .data
    msg1 db 10,13,"Enter 5 64 bit numbers"
    len1 equ $-msg1
    msg2 db 10,13,"Entered 5 64 bit numbers"
    len2 equ $-msg2

section .bss
    array resd 200
    counter resb 1

section .text
    global _start
    _start:

;display
    mov Rax,1
    mov Rdi,1
    mov Rsi,msg1
    mov Rdx,len1
    syscall
```

```
;accept

mov byte[counter],05
mov rbx,00

        loop1:
            mov rax,0                   ; 0 for read
            mov rdi,0                   ; 0 for keyboard
            mov rsi, array              ;move pointer to start of array
            add rsi,rbx
            mov rdx,17
            syscall
                add rbx,17                      ;to move counter
            dec byte[counter]
            JNZ loop1

    ;display
        mov Rax,1
        mov Rdi,1
        mov Rsi,msg2
        mov Rdx,len2
        syscall

    ;display
    mov byte[counter],05
    mov rbx,00
            loop2:
                mov rax,1               ;1 for write
                mov rdi, 1              ;1 for monitor
    mov rsi, array
    add rsi,rbx
    mov rdx,17                  ;16 bit +1 for enter
    syscall
    add rbx,17
    dec byte[counter]
```