```
org 0x7c00        ; Set beginning (where the original IBM machine loads boot)

    ; Print two newlines before the prompt
    mov ah, 0Eh       ; BIOS teletype function
    mov al, 0Ah       ; Newline character
    int 10h           ; Display newline
    mov al, 0Ah       ; Another newline character
    int 10h           ; Display another newline

    ; Display the prompt
    mov si, prompt     ; Point to the string to display

msgloop:
    mov al, [si]       ; Load the current character
    cmp al, 0          ; Check for null terminator
    je endmsgloop      ; If null, end the loop
    int 10h            ; Display the character
    inc si             ; Move to the next character
    jmp msgloop        ; Repeat for next character

endmsgloop:

    ; Print a newline and carriage return before starting to read input
    mov ah, 0Eh       ; BIOS teletype function
    mov al, 0Dh       ; Carriage return character (CR)
    int 10h           ; Display carriage return
    mov al, 0Ah       ; Newline character (LF)
    int 10h           ; Display newline

    ; Now we start reading input from the keyboard
    mov si, buffer     ; Point to the buffer where input will be stored
    xor cx, cx         ; Clear the CX register

read_input:
    mov ah, 0         ; BIOS keyboard input function
    int 16h           ; Wait for a key press
    cmp al, 0Dh       ; Compare with Enter key (CR)
    je end_input      ; If Enter key is pressed, end input

    ;; Echo the character to the screen
    mov ah, 0Eh       ; BIOS teletype function
    int 10h           ;; Print the typed character

    ;; Store in buffer and increment counters
    mov [si], al      ;; Store the character in buffer
    inc si            ;; Move to next buffer position
    inc cx            ;; Increment number of characters typed
    jmp read_input     ;; Repeat for next character

end_input:
   mov byte [si], 0   ;; Null-terminate buffer

   ;; Now send contents of buffer to COM1 (0x3F8)
   mov si, buffer      ;; Point to buffer

send_to_serial:
   mov al, [si]        ;; Load character from buffer
   cmp al, 0           ;; Check for null terminator
   je done_sending      ;; If null, end sending loop
   call send_char_to_com1 ;; Send char to COM1

   inc si               ;; Move to next char in buffer
   jmp send_to_serial   ;; Repeat for next charac

send_char_to_com1:
   ;; Wait for serial port ready (check if Transmit Holding Register is empty)
   mov dx, 0x3F8        ;; COM1 port base address

   out dx, al           ;; Send character in AL to COM1
   ret                   ;; Return from subroutine

done_sending:

prompt db 'Type characters and press <ENTER>: ', 0
buffer db 80          ;; Reserve space for up to 80 characters of input

times 510-($-$$) db 0; Pad with zeros up to 510 bytes
dw 0xaa55             ;; Boot signature
```