DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Subject: Microprocessor and Microcontroller Based Design

SUBMITTED TO:
Dr Saghir Khan

SUBMITTED BY:
Waseem Ghulam (409600)
Ameer Hamza (409662)
Muskan Fatima (414875)
Waqar Ahmed (398417)

Lab 03:String I/O Operations,Flags Register and Control statements

Objectives:

The objective of this lab session is to develop the understanding of string input/output operations using DOS INT 21H function calls, flags register and some control statements.

Tasks:

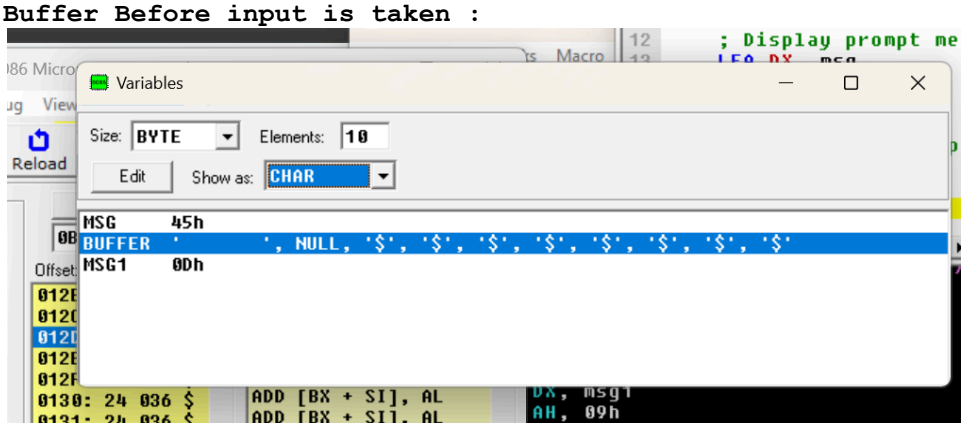
Task1:

Prompt the user for entering his first name. Set the maximum limit to nine characters. Define a buffer of appropriate size and store the input taken from the user in it. Attach screenshots of how the whole buffer looks like before and after the input is taken. Before and after screenshots must be in both hex and char format.

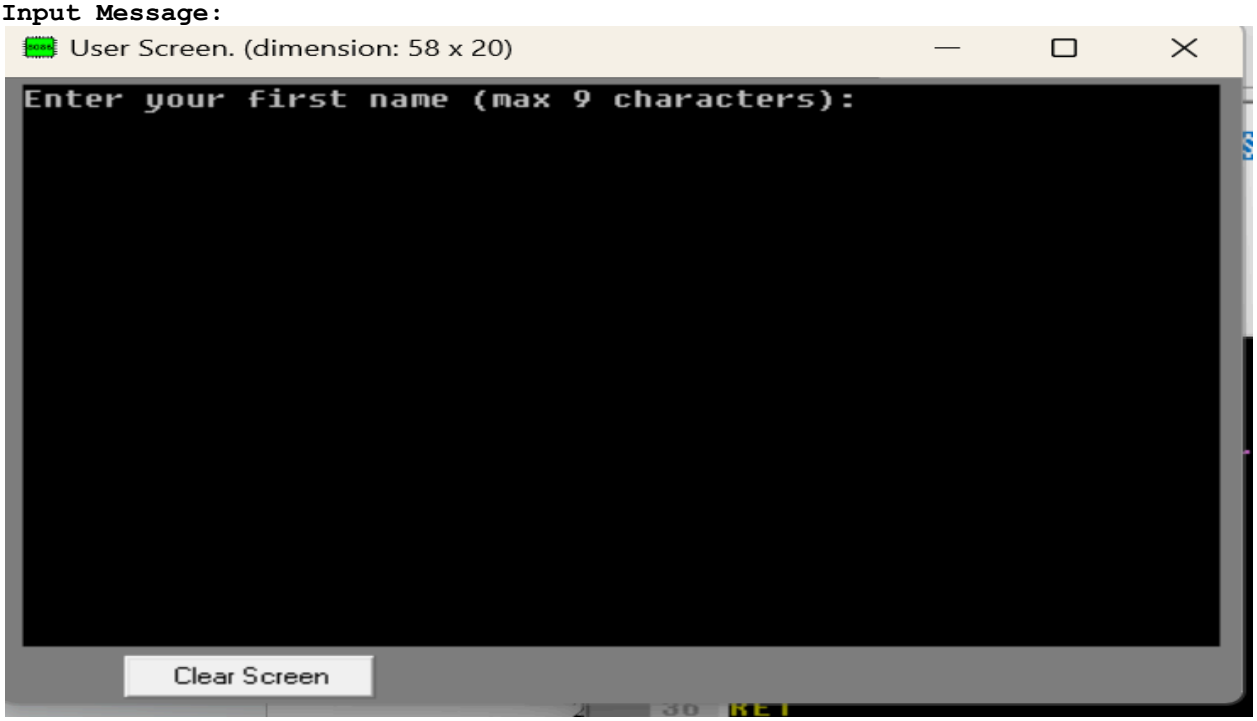
Code:

```
; COM file is loaded at CS:0100h
ORG 100h            ; COM files start execution from offset 100h

.DATA
    msg DB 'Enter your first name (max 9 characters): $'
    buffer DB 9, 0, 9 DUP('$')  ; First byte: max length, Second byte: actual
length, rest: input space
    msg1 DB 0Dh, 0Ah, 'Your name is: $'

.CODE

START:
    ; Display prompt message
    LEA DX, msg
    MOV AH, 09h
    INT 21h

    ; Read the user input (buffered input)
    MOV AH, 0Ah
    LEA DX, buffer
    INT 21h

    ; Display "Your name is:"
    LEA DX, msg1
    MOV AH, 09h
    INT 21h

    ; Display the entered name (from buffer[2] onwards)
    LEA DX, buffer + 2  ; Skip first two bytes (buffer size and input length)
    MOV AH, 09h
    INT 21h

    ; Exit the program
    MOV AH, 4Ch
    INT 21h

RET
```

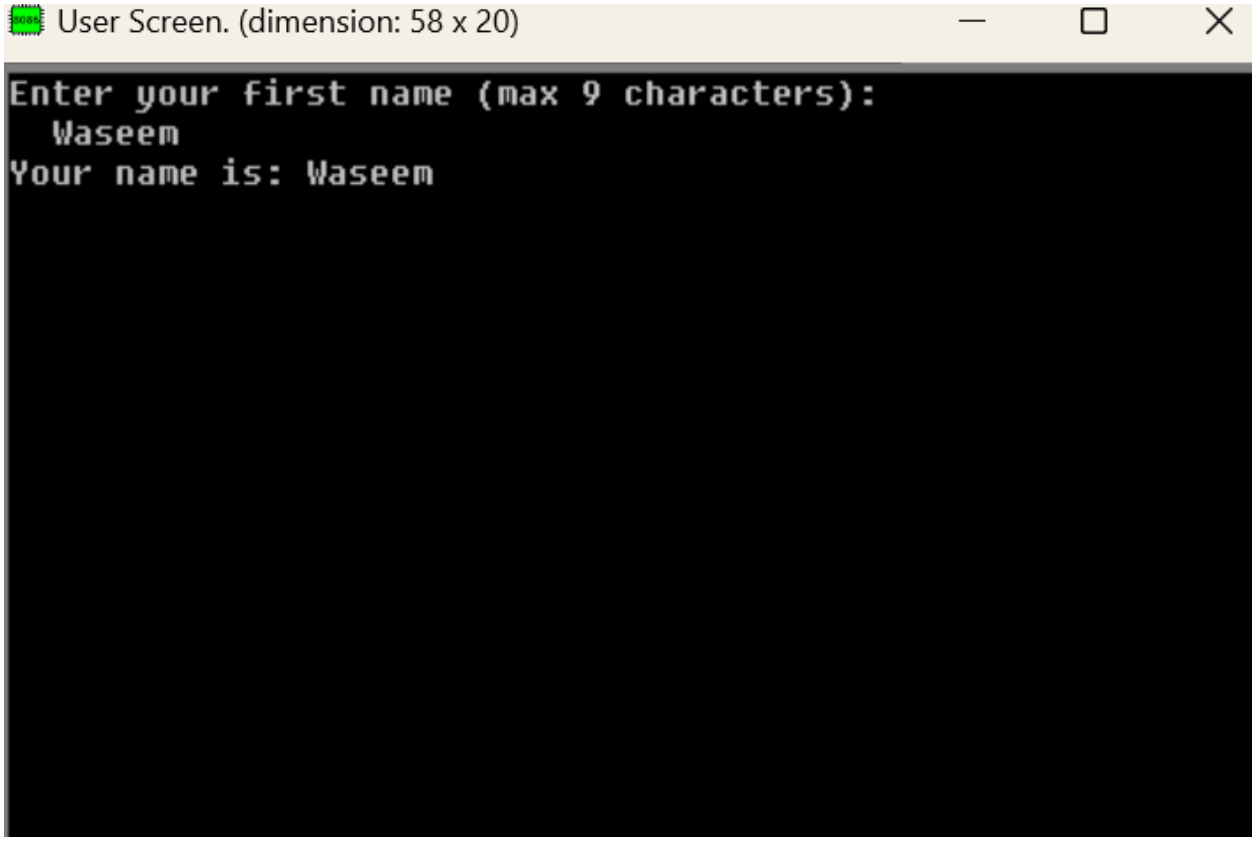
Buffer Before input is taken :



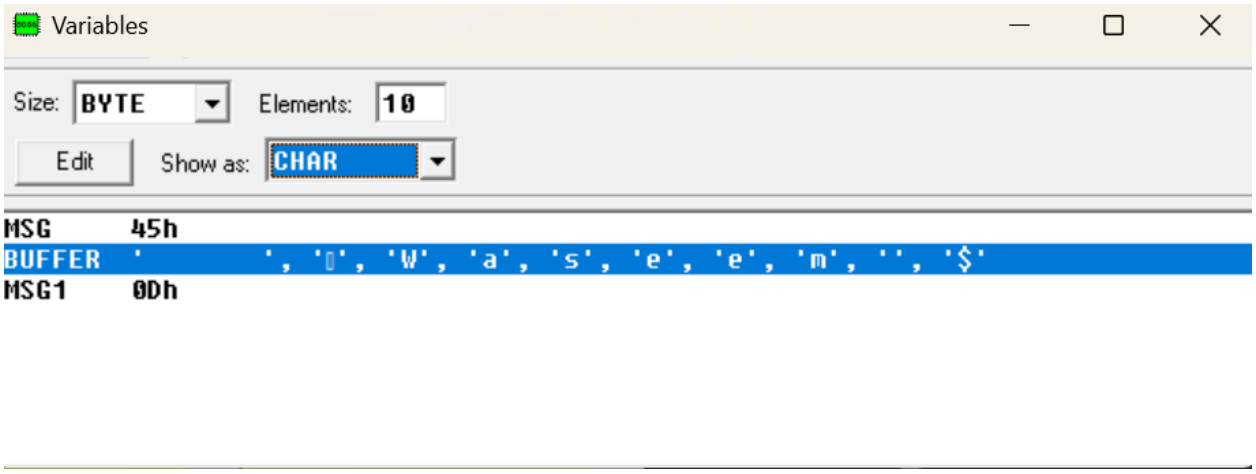
Input Message:



Input and Result:



Buffer After:



Task2:

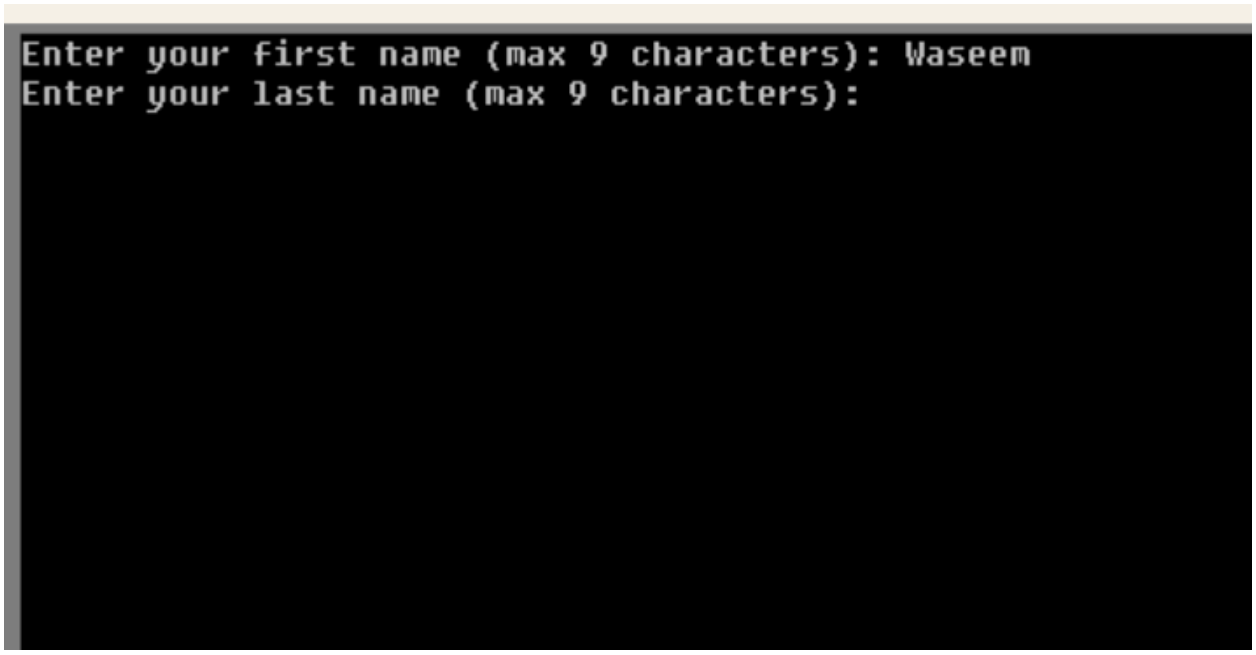
Prompt the user twice to input his first and last name into two separately defined buffers. When both the inputs are received, your code should display the complete name with a space between first and last name. Everything on console should be intuitive and readable.

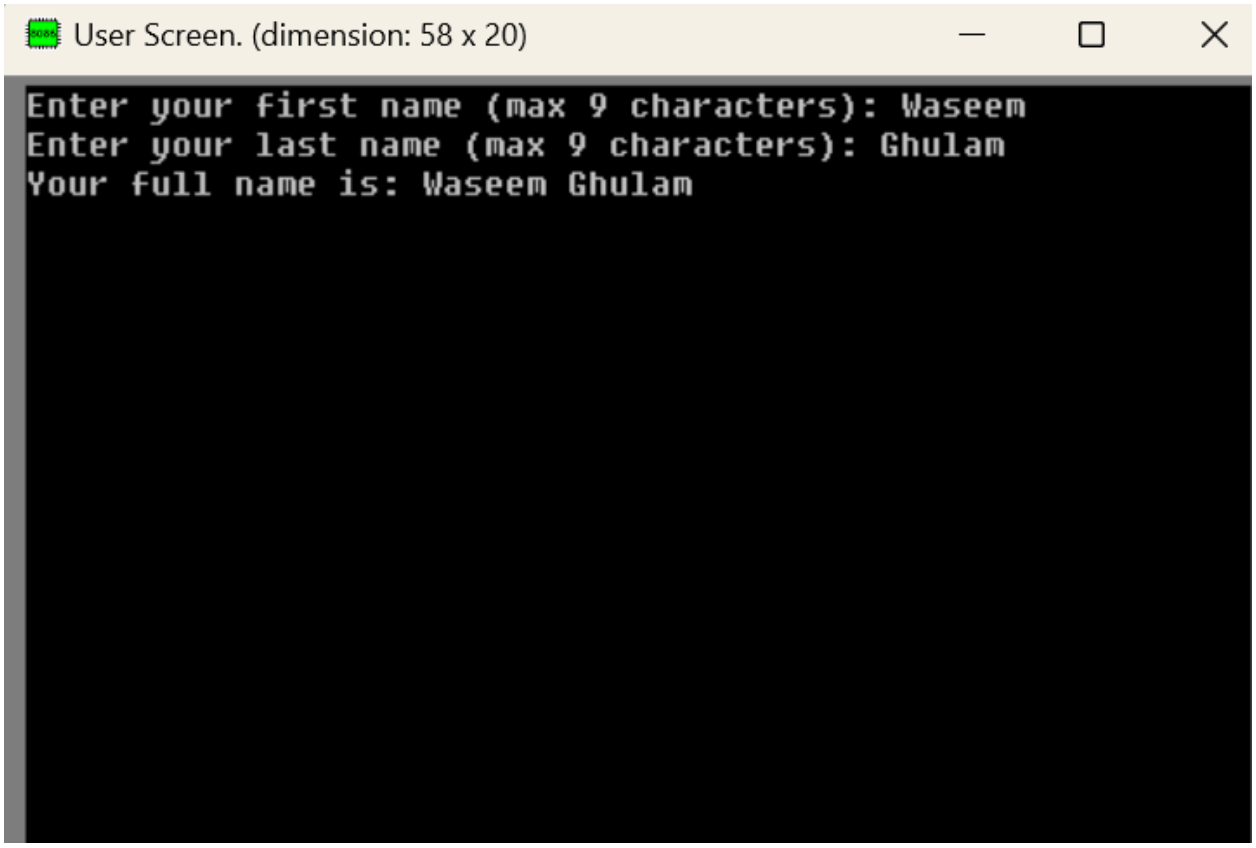
Code:

```
; COM file is loaded at CS:0100h
ORG 100h            ; COM files start execution from offset 100h

.DATA
    msg DB 'Enter your first name (max 9 characters): $'
    msg2 DB 0Dh, 0Ah, 'Enter your last name (max 9 characters): $'
    buffer DB 9, 0, 9 DUP('$')

    buffer2 DB 9, 0, 9 DUP('$')
    msg1 DB 0Dh, 0Ah, 'Your full name is: $'
    space DB ' '

.CODE

START:

    LEA DX, msg
    MOV AH, 09h
    INT 21h
```

```
    MOV AH, 0Ah
    LEA DX, buffer
    INT 21h

    LEA DX, msg2
    MOV AH, 09h
    INT 21h

    MOV AH, 0Ah
    LEA DX, buffer2
    INT 21h

    LEA DX, msg1
    MOV AH, 09h
    INT 21h

    MOV AH, 09H
    MOV BH, 00H
    MOV BL, buffer[1] ;
    MOV buffer[BX+2], '$'
    LEA DX, buffer[2];
    INT 21H

    MOV AH, 02H
    MOV DL, ' '
    INT 21H

    MOV AH, 09H
    MOV BH, 00H
    MOV BL, buffer2[1]  ;
    MOV buffer2[BX+2], '$'
    LEA DX, buffer2[2]; Lo
    INT 21H

    MOV AH, 4Ch
    INT 21h

RET
```

Output:


Enter your first name (max 9 characters): Waseem
Enter your last name (max 9 characters):


User Screen. (dimension: 58 x 20)
Enter your first name (max 9 characters): Waseem
Enter your last name (max 9 characters): Ghulam
Your full name is: Waseem Ghulam

Task3:

Modify the code in the above task so that it can now output the number of characters in first and last name of the user separately. The console window should be intuitive. Paste the code and screenshot of the console.

```
ORG 100h           ; COM files start execution from offset 100h

.DATA
    msg DB 'Enter your first name (max 9 characters): $'
    msg2 DB 0Dh, 0Ah, 'Enter your last name (max 9 characters): $'
    buffer DB 9, 0, 9 DUP('$')

    buffer2 DB 9, 0, 9 DUP('$')
    msg1 DB 0Dh, 0Ah, 'chars in first and last name respectively: $'
    space DB ' '

.CODE

START:

    LEA DX, msg
    MOV AH, 09h
    INT 21h

    MOV AH, 0Ah
    LEA DX, buffer
    INT 21h

    LEA DX, msg2
    MOV AH, 09h
    INT 21h

    MOV AH, 0Ah
    LEA DX, buffer2
    INT 21h

    LEA DX, msg1
    MOV AH, 09h
    INT 21h

    ; Convert number of characters in buffer[1] to ASCII and print
    MOV DL, buffer[1]
    ADD DL, 30h    ; Convert to ASCII
    MOV AH, 02h    ; Print character in DL
    INT 21h

    ; Print a space
    MOV AH, 02h
    MOV DL, ' '
```

```
    INT 21h

    ; Convert number of characters in buffer2[1] to ASCII and print
    MOV DL, buffer2[1]
    ADD DL, 30h    ; Convert to ASCII
    MOV AH, 02h    ; Print character in DL
    INT 21h

    ; Exit the program
    MOV AH, 4Ch
    INT 21h
```

Output: