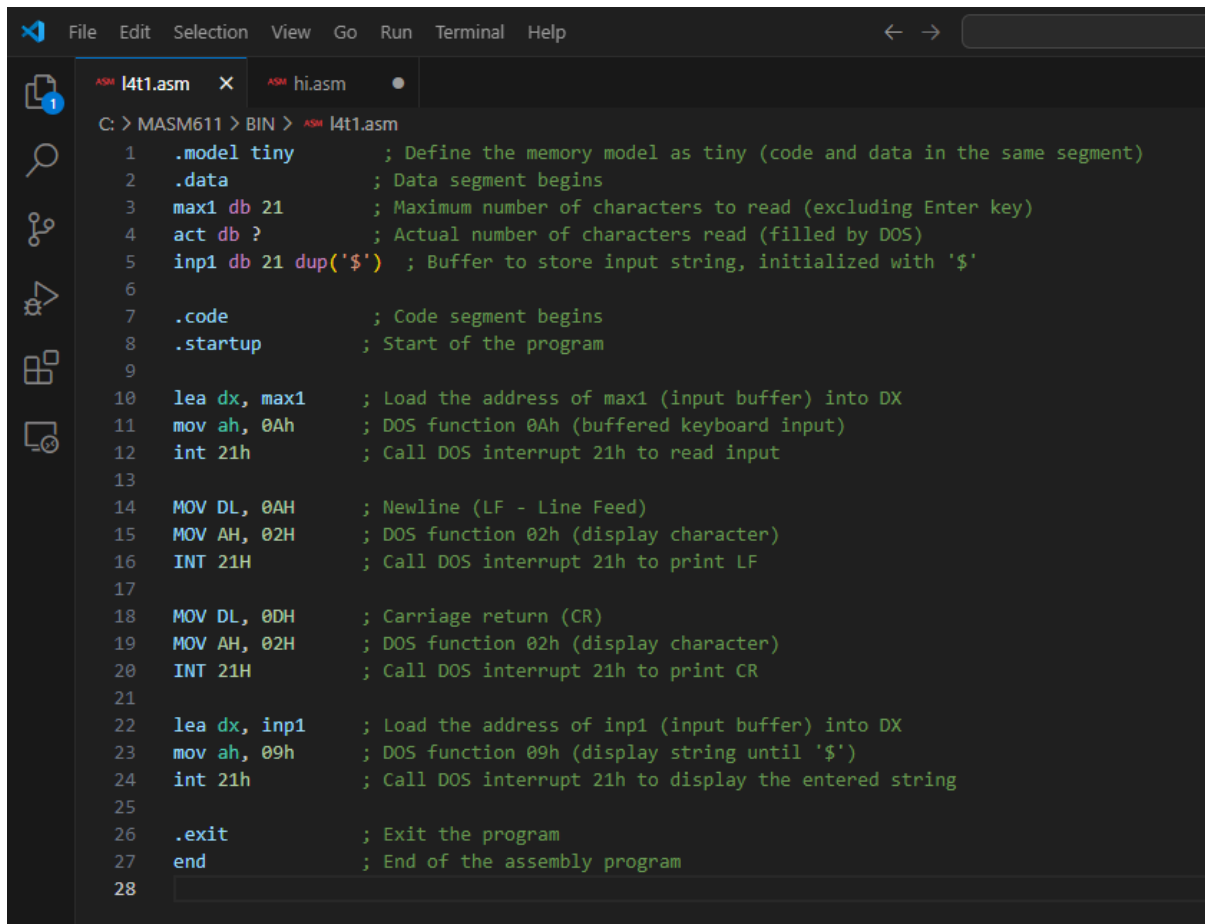


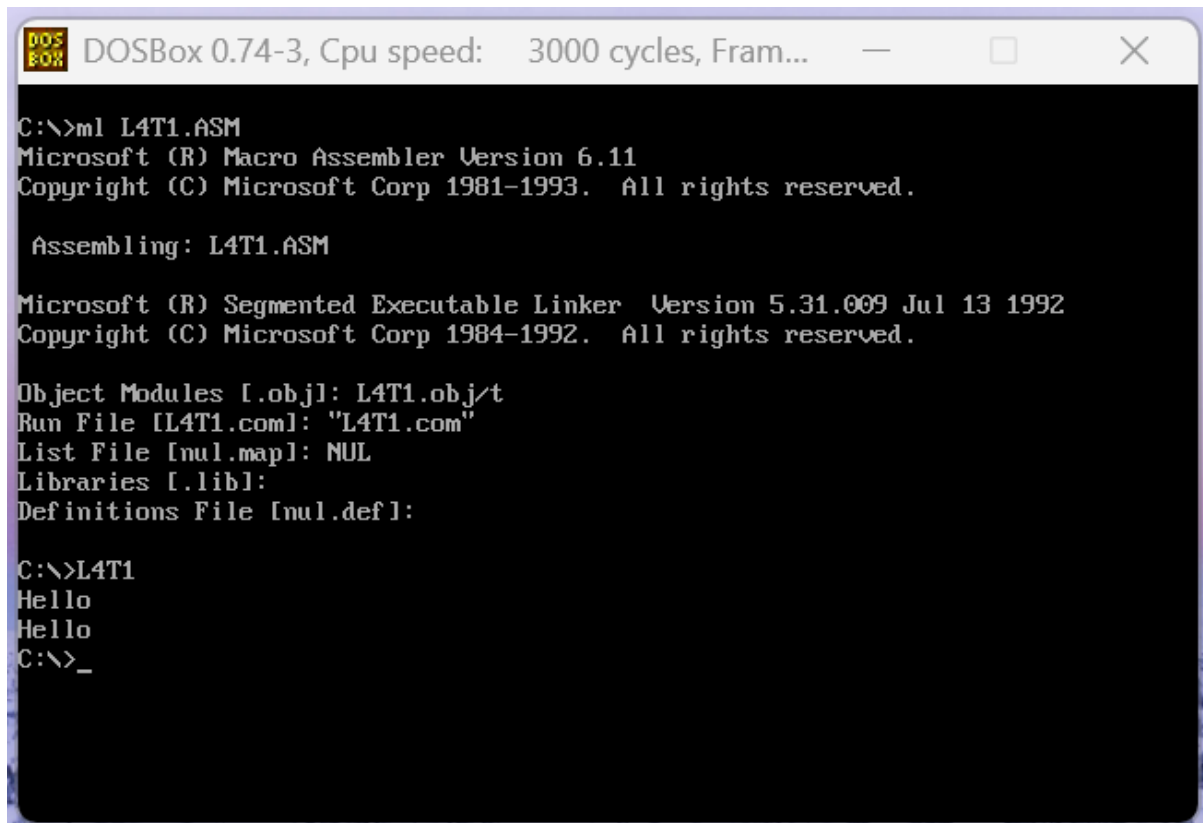
LAB-4 Updated...

Task 1...Code...

A screenshot of a Windows command prompt window. The title bar shows 'File Edit Selection View Go Run Terminal Help'. The command prompt shows the directory 'C:\MASM611\BIN' and the file 'l4t1.asm'. The assembly code is displayed, starting with '.model tiny' and ending with '.exit' and 'end'. The code includes comments for each instruction, such as '; Define the memory model as tiny (code and data in the same segment)' and '; DOS function 0Ah (buffered keyboard input)'.

```
C: > MASM611 > BIN > ASM l4t1.asm
1  .model tiny          ; Define the memory model as tiny (code and data in the same segment)
2  .data                ; Data segment begins
3  max1 db 21           ; Maximum number of characters to read (excluding Enter key)
4  act db ?             ; Actual number of characters read (filled by DOS)
5  inp1 db 21 dup('$')  ; Buffer to store input string, initialized with '$'
6
7  .code                ; Code segment begins
8  .startup              ; Start of the program
9
10 lea dx, max1          ; Load the address of max1 (input buffer) into DX
11 mov ah, 0Ah           ; DOS function 0Ah (buffered keyboard input)
12 int 21h               ; Call DOS interrupt 21h to read input
13
14 MOV DL, 0Ah           ; Newline (LF - Line Feed)
15 MOV AH, 02h           ; DOS function 02h (display character)
16 INT 21h               ; Call DOS interrupt 21h to print LF
17
18 MOV DL, 0Dh           ; Carriage return (CR)
19 MOV AH, 02h           ; DOS function 02h (display character)
20 INT 21h               ; Call DOS interrupt 21h to print CR
21
22 lea dx, inp1          ; Load the address of inp1 (input buffer) into DX
23 mov ah, 09h           ; DOS function 09h (display string until '$')
24 int 21h               ; Call DOS interrupt 21h to display the entered string
25
26 .exit                 ; Exit the program
27 end                   ; End of the assembly program
28
```

Output...



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
C:\>ml L4T1.ASM
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: L4T1.ASM

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

Object Modules [.obj]: L4T1.obj/t
Run File [L4T1.com]: "L4T1.com"
List File [nul.map]: NUL
Libraries [.lib]:
Definitions File [nul.def]:

C:\>L4T1
Hello
Hello
C:\>_
```

Task2... Code...

```
.model tiny          ; Define the memory model as tiny (code and data in the same
segment)
.data                ; Data segment begins

dat1 db 'Enter user name$' ; Prompt message for username input (terminated by
'$')
pred db 'Hello '          ; Greeting message prefix (message shown after
successful login)
name1 db 'mritunjay$'      ; Predefined username (stored for comparison)
password db '1234'         ; Predefined password (stored for comparison)

max1 db 10               ; Max characters for username input (maximum input
length allowed)
act db ?                 ; Actual number of characters read (filled by DOS
after input)
inp1 db 10 dup('$')       ; Buffer for username input (initialized with '$'
for termination)

max2 db 6                ; Max characters for password input (maximum input
length allowed)
act2 db ?                ; Actual number of characters read for password
```

```

inp2 db 6 dup('$')          ; Buffer for password input (initialized with '$'
for termination)

inp3 db 'Enter password $'  ; Prompt message for password input (terminated
by '$')

.code                      ; Code segment begins
.startup                  ; Start of the program

; Display "Enter user name" prompt
lea dx, dat1              ; Load address of dat1 (username prompt) into DX
mov ah, 09h               ; DOS function 09h (display string terminated by '$')
int 21h                   ; Call interrupt 21h to display the string

; Print newline (LF) and carriage return (CR) for better output formatting
MOV DL, 0AH               ; Line Feed (LF) character
MOV AH, 02H               ; DOS function 02h (print single character)
INT 21H                   ; Print LF (new line)
MOV DL, 0DH               ; Carriage Return (CR) character
MOV AH, 02H               ; DOS function 02h (print single character)
INT 21H                   ; Print CR (move cursor to beginning of line)

; Read username input from the user
lea dx, max1              ; Load address of max1 (buffer size for username) into DX
mov ah, 0Ah               ; DOS function 0Ah (buffered input)
int 21h                   ; Call interrupt 21h to read the input from user

; Print newline (LF) and carriage return (CR) again after reading input
MOV DL, 0AH               ; Line Feed (LF)
MOV AH, 02H               ; DOS function 02h
INT 21H                   ; Print LF
MOV DL, 0DH               ; Carriage Return (CR)
MOV AH, 02H               ; DOS function 02h
INT 21H                   ; Print CR

; Compare the entered username with the predefined username
mov cx, 8                 ; Set loop counter to 8 (maximum characters in username)
lea si, name1             ; Load address of predefined username (name1) into SI
lea di, inp1              ; Load address of user input (inp1) into DI

l2:
mov al, [si]              ; Load character from predefined username into AL
cmp al, [di]              ; Compare the character with the user input
jne l1                    ; If they don't match, jump to l1 (exit)
dec cl                    ; Decrease loop counter
jnz l2                    ; If counter is not zero, continue comparison

; Display "Enter password" prompt after successful username match

```

```

lea dx, inp3    ; Load address of password prompt into DX
mov ah, 09h     ; DOS function 09h
int 21h        ; Call interrupt 21h to display the string

; Print newline (LF) and carriage return (CR) again for output formatting
MOV DL, 0AH     ; Line Feed (LF)
MOV AH, 02H     ; DOS function 02h
INT 21H        ; Print LF
MOV DL, 0DH     ; Carriage Return (CR)
MOV AH, 02H     ; DOS function 02h
INT 21H        ; Print CR

; Read password input from the user
lea dx, max2    ; Load address of max2 (buffer size for password) into DX
mov ah, 0Ah     ; DOS function 0Ah (buffered input)
int 21h        ; Call interrupt 21h to read the input from user

; Compare the entered password with the predefined password
mov cx, 5       ; Set loop counter to 5 (maximum characters in password)
lea si, password ; Load address of predefined password into SI
lea di, inp2     ; Load address of user input into DI

l3:
mov al, [si]    ; Load character from predefined password into AL
cmp al, [di]    ; Compare the character with the user input
jne l1          ; If they don't match, jump to l1 (exit)
dec cl         ; Decrease loop counter
jnz l3         ; If counter is not zero, continue comparison

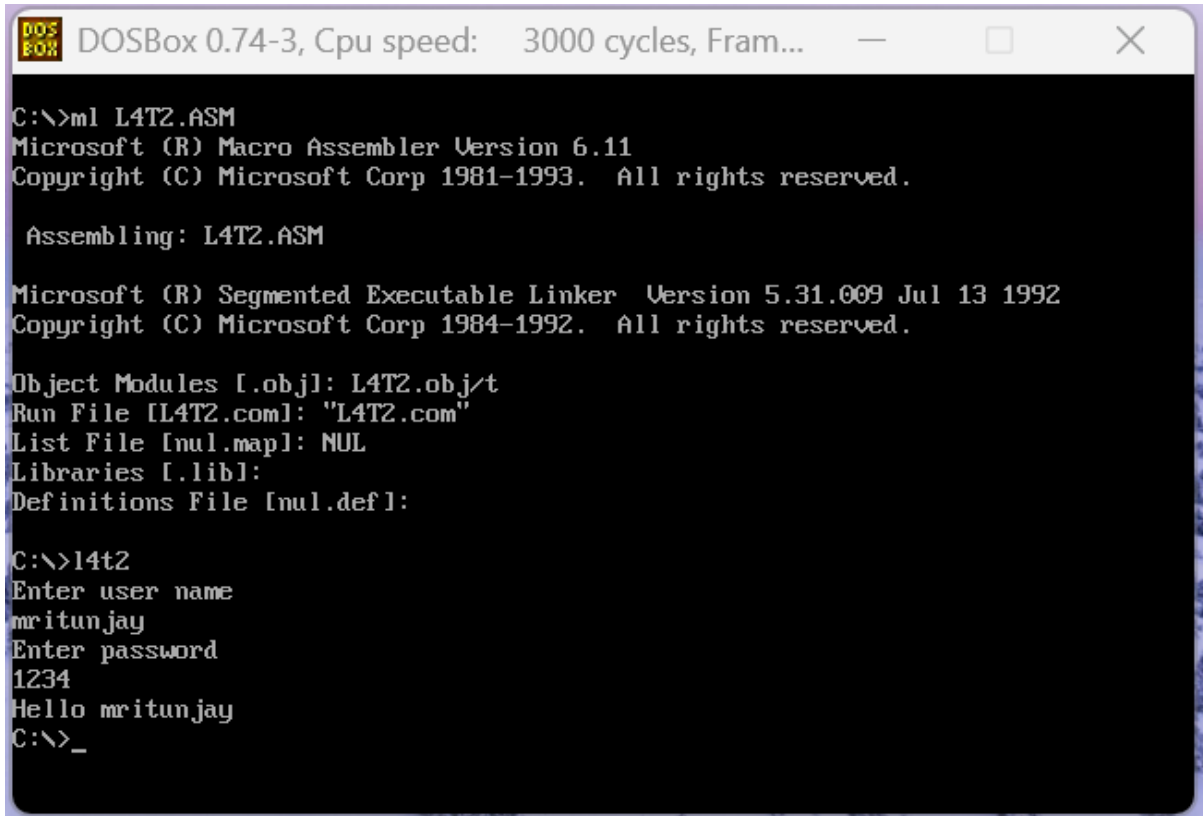
; Print newline (LF) and carriage return (CR) after password input
MOV DL, 0AH     ; Line Feed (LF)
MOV AH, 02H     ; DOS function 02h
INT 21H        ; Print LF
MOV DL, 0DH     ; Carriage Return (CR)
MOV AH, 02H     ; DOS function 02h
INT 21H        ; Print CR

; Display greeting message "Hello " after successful username and password
match
lea dx, pred    ; Load address of greeting message into DX
mov ah, 09h     ; DOS function 09h (display string terminated by '$')
int 21h        ; Call interrupt 21h to display the string

l1:
.EXIT          ; Exit the program
end            ; End of the assembly program

```

Output...



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
C:\>ml L4T2.ASM
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: L4T2.ASM

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

Object Modules [.obj]: L4T2.obj/t
Run File [L4T2.com]: "L4T2.com"
List File [nul.map]: NUL
Libraries [.lib]:
Definitions File [nul.def]:

C:\>l4t2
Enter user name
mritunjay
Enter password
1234
Hello mritunjay
C:\>_
```

Task3... Code...

```
.model tiny          ; Define the memory model as tiny (code and data in the same
segment)
.data                ; Data segment begins

dat1 db 'Enter user name$' ; Prompt message for username input
pred db 'Hello '          ; Greeting message prefix
name1 db 'mritunjay$'      ; Predefined username
password db '98765'        ; Predefined password

max1 db 10               ; Max characters for username input
act db ?                 ; Actual number of characters read (filled by DOS)
inp1 db 10 dup('$')       ; Buffer for username input

max2 db 6                ; Max characters for password input
act2 db ?                ; Actual number of characters read (filled by DOS)
inp2 db 6 dup('$')        ; Buffer for password input
```

```

inp3 db 'Enter password $' ; Prompt message for password input

.code          ; Code segment begins
.startup       ; Start of the program

; Display "Enter user name"
lea dx, dat1
mov ah, 09h
int 21h

; Print newline (LF) and carriage return (CR)
MOV DL, 0AH
MOV AH, 02H
INT 21H
MOV DL, 0DH
MOV AH, 02H
INT 21H

; Read username input
lea dx, max1
mov ah, 0Ah
int 21h

; Print newline (LF) and carriage return (CR)
MOV DL, 0AH
MOV AH, 02H
INT 21H
MOV DL, 0DH
MOV AH, 02H
INT 21H

; Compare entered username with predefined username
mov cx, 8      ; Set loop counter (8 characters max)
lea si, name1  ; Load predefined username into SI
lea di, inp1   ; Load user input into DI

l2:
mov al, [si]   ; Load character from predefined username
cmp al, [di]   ; Compare it with user input
jne l1         ; If not equal, jump to l1 (exit)
dec cl        ; Decrease counter
jnz l2         ; Loop until counter reaches zero

; Display "Enter password"
lea dx, inp3
mov ah, 09h
int 21h

```

```

; Print newline (LF) and carriage return (CR)
MOV DL, 0AH
MOV AH, 02H
INT 21H
MOV DL, 0DH
MOV AH, 02H
INT 21H

; Read password input with '*' masking
lea di, inp2    ; Load password buffer into DI
mov cx, 5       ; Set loop counter (5 characters)

l4:
mov ah, 08      ; DOS function 08h (read single character without echo)
int 21h
mov [di], al    ; Store input character in password buffer

mov dl, '*'     ; Display '*' instead of actual input
mov ah, 02h     ; DOS function 02h (display character)
int 21h

inc di          ; Move to next character in buffer
dec cx          ; Decrease counter
jnz l4          ; Repeat until all 5 characters are read

; Compare entered password with predefined password
mov cx, 5       ; Set loop counter (5 characters)
lea si, password ; Load predefined password into SI
lea di, inp2    ; Load user input into DI

l3:
mov al, [si]    ; Load character from predefined password
cmp al, [di]    ; Compare with user input
jne l1          ; If not equal, jump to l1 (exit)
dec cl          ; Decrease counter
jnz l3          ; Loop until counter reaches zero

; Print newline (LF) and carriage return (CR)
MOV DL, 0AH
MOV AH, 02H
INT 21H
MOV DL, 0DH
MOV AH, 02H
INT 21H

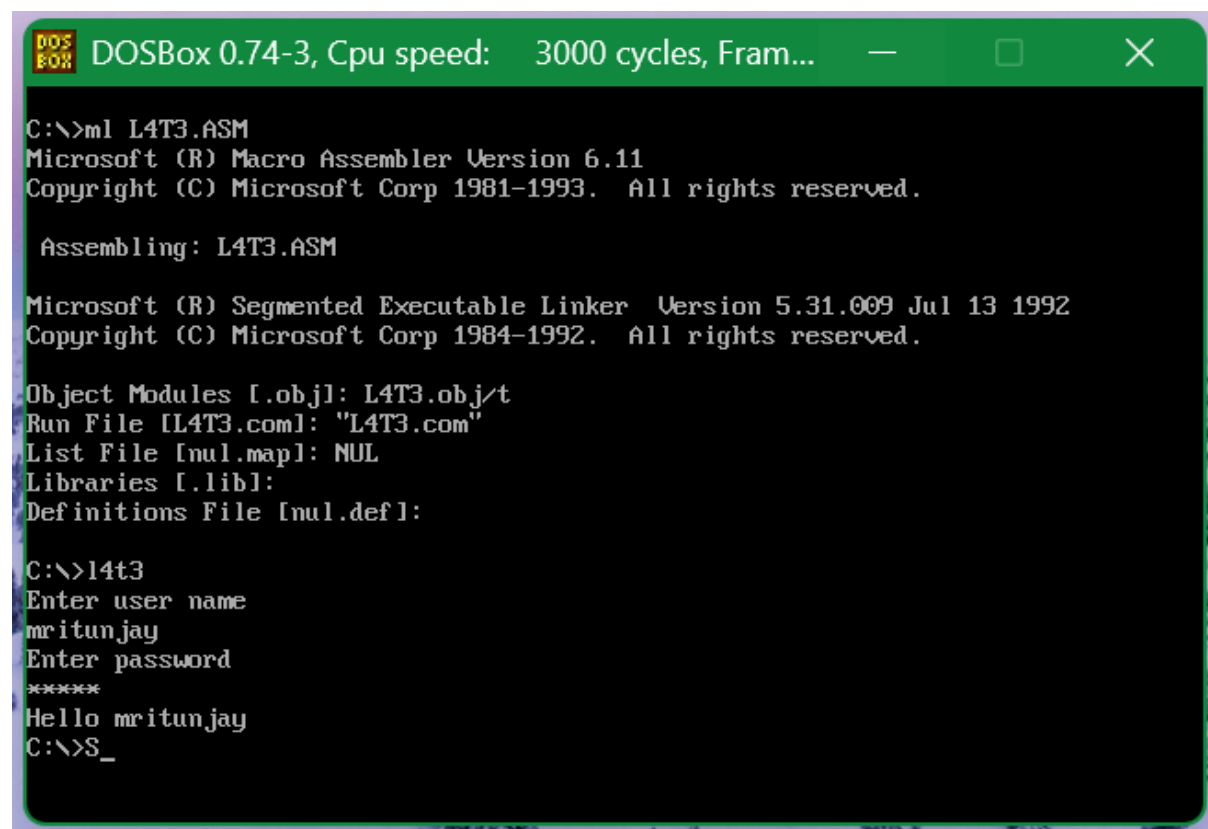
; Display greeting message "Hello "
lea dx, pred

```

```
mov ah, 09h
int 21h

l1:
.exit          ; Exit the program
end            ; End of the assembly program
```

Output...



The screenshot shows a DOSBox 0.74-3 window with a green title bar. The command prompt inside shows the following sequence of commands and output:

```
C:\>ml L4T3.ASM
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: L4T3.ASM

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

Object Modules [ .obj ]: L4T3.obj/t
Run File [L4T3.com]: "L4T3.com"
List File [nul.map]: NUL
Libraries [ .lib ]:
Definitions File [nul.def]:

C:\>l4t3
Enter user name
mritunjay
Enter password
*****
Hello mritunjay
C:\>S_
```