Name : Sajid Islam

Instructor : Muhammad Hassaan

COAL

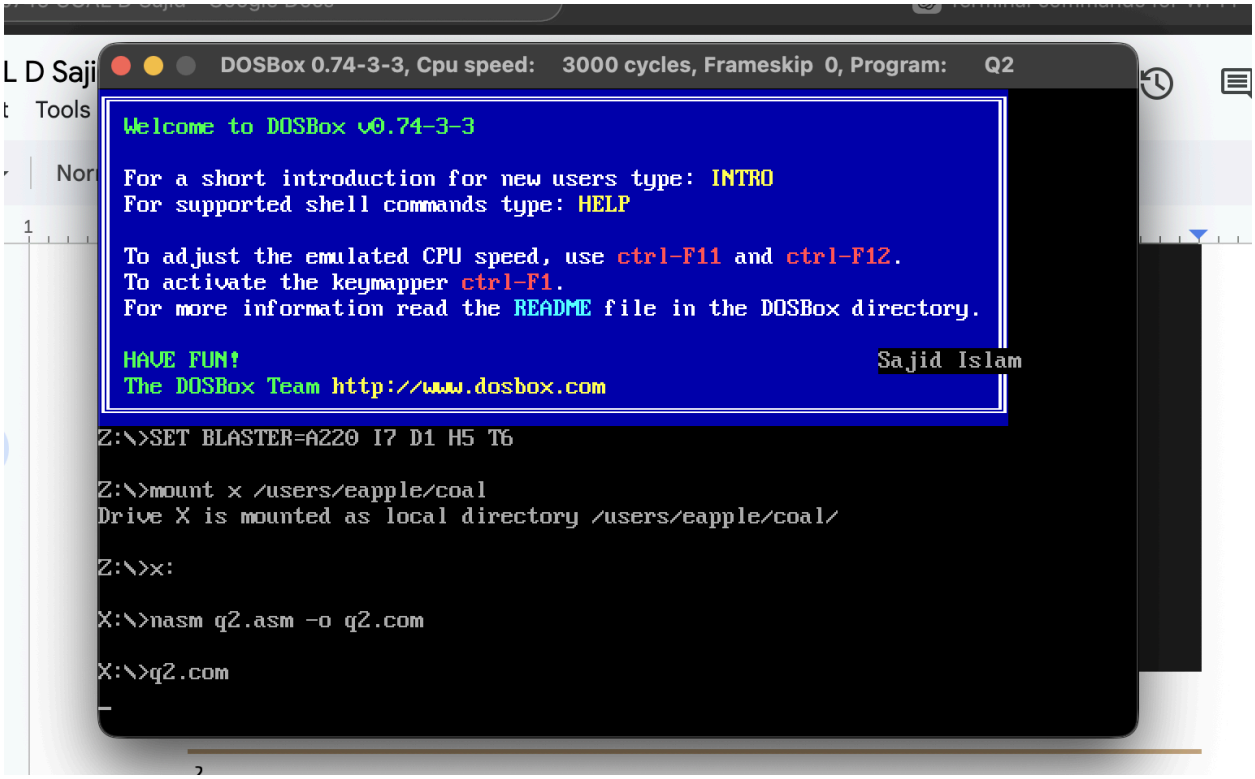
Assessment No 12

Roll No : 24p-0745

Q1

```
[org 0x0100]

jmp start

name:   db "Sajid Islam", 0   ; string to print, ending with 0
row:    db 10                 ; row number
col:    db 60                 ; column number


print_string:
    push bp              ; save old base pointer
    mov bp, sp           ; set new stack frame

    push ax              ; save registers
    push di
    push si

    mov di, [bp + 6]     ; load screen position
    mov si, [bp + 4]     ; load string address

    mov ax, 0xb800       ; video memory segment
    mov es, ax
    mov ah, 0x07         ; attribute (light grey on black)

; Loop to print each character
looop:
        mov al, [si]         ; get next character
        mov word [es:di], ax ; write char + attribute to screen
        add di, 2            ; move to next screen cell
        add si, 1            ; move to next character in string
        cmp byte [si], 0     ; check if string ended
        jnz looop            ; if not 0, keep printing

    pop si               ; restore registers
    pop di
    pop ax
    pop bp               ; restore BP
    ret 4                ; remove 4 bytes of parameters and return

start:
    mov ax , 80                ; 80 columns
    mul byte [row]             ; ax = row * 80
    add al, byte [col]         ; add column number
    shl ax, 1                  ; each cell = 2 bytes → multiply by 2

    push ax                    ; push screen position
    push name                  ; push string address
    call print_string          ; print the name

    mov ax, 0x0000
    int 0x16

    mov ah ,0x4c
    int 0x21
```

OUTPUT


DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: Q2
Welcome to DOSBox v0.74-3-3
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Sajid Islam
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount x /users/eapple/coal
Drive X is mounted as local directory /users/eapple/coal/
Z:\>x:
X:\>nasm q2.asm -o q2.com
X:\>q2.com

Q2

```
[org 0x0100]

jmp start

num:    dw 0x1A3F        ; 16-bit number
row:    db 0              ; top row
col:    db 5              ; starting column

start:
    mov ax, 80
    mul byte [row]       ; row * 80
    add ax, word [col]   ; add column
    shl ax, 1            ; each cell = 2 bytes
    mov di, ax

    mov ax, [num]        ; load number
    mov cx, 0
    mov bx, 16           ; hex base

convert_hex:
    xor dx, dx
    div bx               ; divide by 16
    push dx              ; store remainder
    inc cx
    cmp ax, 0
    jne convert_hex

    mov ax, 0xB800
    mov es, ax

print_loop:
    pop dx
    cmp dx, 10
    jl digit_is_number
    add dl, 'A' - 10     ; convert 10-15 to 'A'-'F'
    jmp store_digit

digit_is_number:
    add dl, '0'          ; convert 0-9 to '0'-'9'

store_digit:
    mov ah, 0x1E         ; color attribute
    mov [es:di], ax
    add di, 2
    dec cx
    jnz print_loop

    xor ax, ax
    int 0x16             ; wait for key press

    mov ah, 0x4C
    int 0x21             ; exit program
```

Q3

```
[org 0x0100]

jmp start

r_start:    dw  5       ; row number
c_start:    dw  10      ; column number
box_width:  dw  30
box_height: dw  8

print_row:
    push bp
    mov bp, sp

    push ax
    push cx

    mov di, [bp + 6]
    mov cx, [bp + 4]
    mov ax, '#'
    mov ah, 0x07

    rep stosw

    pop cx
    pop ax
    pop bp
    ret 4

print_col:
    push bp
    mov bp, sp

    push ax
    push cx

    mov di, [bp + 6]
    mov cx, [bp + 4]
    mov ax, '#'
    mov ah, 0x07
```

```
row_loop:
    mov word [es:di], ax
    add di, 160
    dec cx
    jnz row_loop

    pop cx
    pop ax
    pop bp
    ret 4

start:
    mov ax, 0xb800
    mov es, ax

cls:
    mov cx, 2000
    mov di, 0
    mov ax, 0x0720
    rep stosw

    ; --- Top horizontal line ---
    mov ax, 80
    mul word [r_start]
    add ax, word [c_start]
    shl ax, 1

    push ax
    push word [box_width]
    call print_row

    ; --- Left vertical line ---
    push di
    push word [box_height]
    call print_col

    ; --- Right vertical line ---
    push ax
    push word [box_height]
    call print_col
```

```
; --- Top horizontal line ---
mov ax, 80
mul word [r_start]
add ax, word [c_start]
shl ax, 1

push ax
push word [box_width]
call print_row

; --- Left vertical line ---
push di
push word [box_height]
call print_col

; --- Right vertical line ---
push ax
push word [box_height]
call print_col

; --- Bottom horizontal line ---
sub di, 160
push di
push word [box_width]
call print_row

; wait for key press
mov ax, 0x0000
int 0x16

; exit
mov ax, 0x4c00
int 0x21
```

Output


DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: Q4