



## Lab 10

### Computer Organization and Assembly Language

<b>Student Name</b>	Umamah Hussain
<b>Registration #</b>	21I-1858
<b>Instructor</b>	Hazoor Ahmad
<b>Class</b>	CS3
<b>Section</b>	A1
<b>Semester</b>	Fall 2022

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

## Assembly Language Code

```
[[org 0x100]
jmp start

msg1: db 'Hi! I am Umamah',0
msg2: db 'I am happy',0
msg3: db 'I Study at FAST.',0
msg4: db 'My Roll No is 21I-1858',0
```

clrscr:

```
    push ax
    push es
    push cx
    push di

    mov ax,0xb800
    mov es,ax
    xor di,di
    mov ax,0x0720
    mov cx,2000
    rep stosw

    pop di
    pop cx
    pop es
    pop ax
```

```
strlen: push bp
        mov bp,sp
        push es
        push cx
        push di
        les di, [bp+4]
        mov cx, 0xffff
        xor al, al
        repne scasb
        mov ax, 0xffff
        sub ax, cx
        dec ax
        pop di
```

```
pop cx
pop es
pop bp
ret 4
```

```
printstr: push bp
mov bp, sp
push es
push ax
push cx
push si
push di
push ds
mov ax, [bp+4]
push ax
call strlen
cmp ax, 0
jz exit
mov cx, ax
mov ax, 0xb800
mov es, ax
mov al, 80
mul byte [bp+8]
add ax, [bp+10]
shl ax, 1
mov di, ax
mov si, [bp+4]
mov ah, [bp+6]
cld
nextchar: lodsb
stosw
loop nextchar
exit: pop di
pop si
pop cx
pop ax
pop es
pop bp
ret 8
```

```
;
```

```
start: mov ah, 0x10
mov al, 03
mov bl, 01
int 0x10
mov ah, 0
int 0x16
call clrscr
```

```
mov ah, 0
int 0x16
mov ax, 20
push ax
mov ax, 1
push ax
mov ax, 02
push ax
mov ax, msg1
push ax
call printstr
```

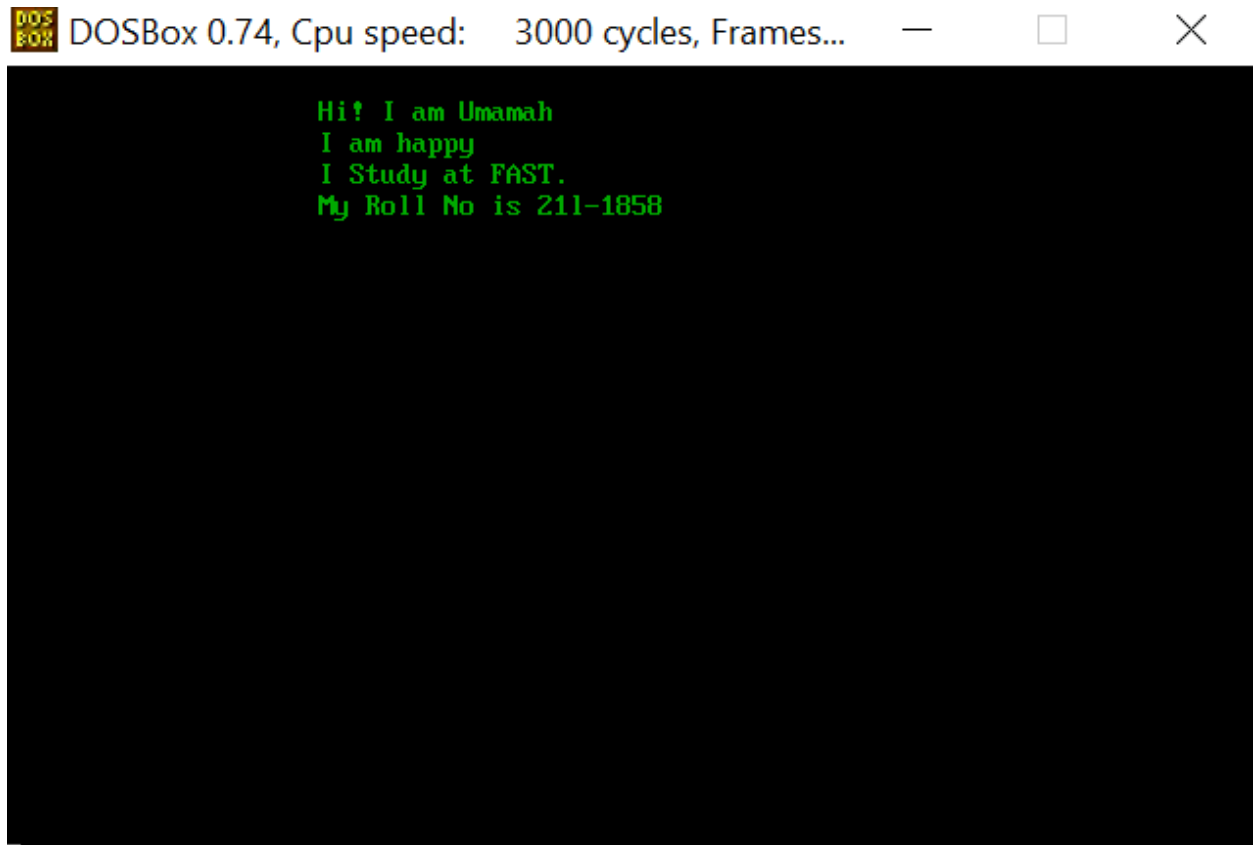
```
mov ah, 0
int 0x16
mov ax, 20
push ax
mov ax, 2
push ax
mov ax, 02
push ax
mov ax, msg2
push ax
call printstr
```

```
mov ah, 0
int 0x16
mov ax, 20
push ax
mov ax, 3
push ax
mov ax, 02
push ax
mov ax, msg3
push ax
call printstr
```

```
mov ah, 0
int 0x16
mov ax, 20
push ax
mov ax, 4
push ax
mov ax, 02
push ax
mov ax, msg4
push ax
call printstr
```

```
mov ah, 0  
int 0x16  
  
mov ax, 0x4c00  
int 0x21
```

## Debugging Screenshots



# Activity 2

## Assembly Language Code

```
[org 0x0100]
jmp code
w equ 50 ; width offset
x equ 50 ; starting x coordinate of line
y equ 100 ; starting y coordinate of line
c equ 60 ; color
a equ 150
b equ 100
e equ 100
d equ 50
```

```
code: mov ah, 0
mov al, 13h
int 10h
; draw diagonal 11:
mov cx, x
mov dx, y
mov al, c
u1: inc dx
mov ah, 0ch ; put pixel
int 10h
inc cx
cmp cx, x+w
jbe u1
; draw diagonal 12:
```

```
mov cx, b
mov dx, x
mov al, c
u2: inc dx
mov ah, 0ch ; put pixel
int 10h
dec cx
cmp cx, 50
jge u2
```

```
; draw diagonal 11:
mov cx, a
mov dx, b
mov al, c
u3: inc dx
mov ah, 0ch ; put pixel
int 10h
```

```
dec cx
cmp cx, a-w
jge u3
; draw diagonal 11:
mov cx, e
mov dx, d
mov al, c
u4: inc dx
mov ah, 0ch ; put pixel
int 10h
inc cx
cmp cx, e+w
jbe u4
;wait for keypress
mov ah,00
int 16h
mov ax, 0x4c00
int 21h
```

## Debugging Screenshots

