|  |
| --- |
| FAST National University |
| **QUIZ 2** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Moeez Ali |
| **Registration #** | 21I-1798 |
| **Instructor** | Hazoor Ahmad |
| **Class** | CS3 |
| **Section** | 3 D |
| **Semester** | Fall 2022 |

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

## **Assembly Language Code**

[org 0x0100]

jmp start

*;//=====================================================================*

*;//*

*;// ###### ##### ## ### ## ## #### ## #####*

*;// ## ## ## ## ## ## #### ## ## ## ##*

*;// ## ##### ## ## ## ## ## ## ## ### ## #####*

*;// ## ## ## ## ####### ## ### ## ## ## ##*

*;// ## ## ## ## ## ## ## ## #### ###### #####*

*;//*

*;//=====================================================================*

triangle:

push bp

mov bp, sp

push ax

push bx

push cx

push di

push es

call clrscr

mov ax, 0xb800

mov es, ax

mov ax, 80

mul byte [bp + 16]

add ax, [bp + 18]

shl ax, 1

mov di, ax

mov cx, [bp + 14]

mov ah, [bp + 6]

mov al, [bp + 4]

loop1 : mov [es:di], ax

add di, 160

loop loop1

mov cx, [bp + 10]

mov dx, [bp + 14]

sub cx, dx

loop2: mov [es:di], ax

add di, 2

loop loop2

mov cx, [bp + 10]

mov dx, [bp + 14]

sub cx, dx

loop3: mov [es:di], ax

sub di, 162

loop loop3

pop es

pop di

pop cx

pop bx

pop ax

pop bp

ret 16

*;//========================================================*

*;//*

*;// #### ## ##### ### #####*

*;// ## ## ## ## ## ## ##*

*;// ## ## ##### ## ## #####*

*;// ## ## ## ####### ## ##*

*;// #### ###### ##### ## ## ## ##*

*;//*

*;//========================================================*

clrscr:

push es

push ax

push bx

push cx

push dx

push si

push di

mov ax, 0xb800

mov es, ax

xor di, di

mov ax, 0x0720

mov cx, 2000

cld

rep stosw

pop di

pop si

pop dx

pop cx

pop bx

pop ax

pop es

ret

start:

*;//==============================================================*

*;//*

*;// ## ## ##### ##### ###### ## #### ##### ####*

*;// ## ## ## ## ## ## ## ## ## ##*

*;// ## ## ##### ##### ## ## ## ##### ###*

*;// ## ## ## ## ## ## ## ## ## ##*

*;// ### ##### ## ## ## ## #### ##### ####*

*;//*

*;//==============================================================*

*;X1*

mov ax, 45

push ax

*;Y1*

mov bx, 10

push bx

*;X2*

mov ax, 10

push ax

*;Y2*

mov bx, 10

push bx

*;X3*

mov ax, 20

push ax

*;Y3*

mov bx, 20

push bx

*;COLOR*

mov ax, 0x03

push ax

*;CHARACTER*

mov bx, '&'

push bx

call triangle

mov ax, 0x4c00

int 21h

## **Debugging Screenshots**



# Activity 2

## **Assembly Language Code**

[org 0x100]

jmp start

*;//==================================================*

*;//*

*;// #### ##### ##### ## ##*

*;// ## ## ## ## ## ## ##*

*;// ## ## ## ##### ####*

*;// ## ## ## ## ##*

*;// #### ##### ## ##*

*;//*

*;//==================================================*

copy\_paste:

push bp

mov bp,sp

push es

push ax

push bx

push cx

push dx

push di

push si

mov ax,0xb800

mov es,ax

xor di,di

mov si,2880

loop1:

mov word ax,[es:di]

mov word [es: si],ax

add di,2

add si,2

cmp si,4000

jle loop1

pop si

pop di

pop dx

pop cx

pop bx

pop ax

pop es

pop bp

ret

start:

call copy\_paste

mov ax,0x4c00

int 21h

## **Debugging Screenshots**

