7.

a) Read your name from the keyboard and display it at a specified location on the screen after

the message “What is your name?” You must clear the entire screen before display.

```
.model small

.stack

.data

.code

m db "what is ur name? $"

s db 30 dup(00h)

l db 00h

row db 0ch

col db 27h

mov ax,@data

mov ds,ax

lea dx,m

mov ah,09h

int 21h

lea si,s

up1: mov ah,01h

 int 21h

 cmp al,0dh

 je down1

 mov [si],al

 inc si

 inc l

 jmp up1
```

```
down1: mov ah,06h
 mov ch,00h
 mov cl,00h
 mov dh,32h
 mov dl,4fh
 mov bh,07h
 mov al,00h
 int 10h
 mov ah,02h
 mov bh,00h
 mov dh,row
 mov dl,col
 int 10h
 lea dx,m
 mov ah,09h
 int 21h
 lea si,s
 mov cl,l
 mov ch,00h
up2: mov dl,[si]
 mov ah,02h
 int 21h
 inc si
 loop up2
 mov ah,4ch
 int 21h
 end
```

b) Scan an 8 x 3 keypad for key closure and to store the code of the key pressed in a memory

location or display on screen. Also display row and column numbers of the key pressed.

```
.model small

.stack

.data

.code

array db “0123456789.+–*/%zykecrnm”

m db 10, 13,”Enter the KEY : $”

m1 db 10, 13,”KEY pressed is : $”

m2 db 10, 13,”Row NO : $”

m3 db 10, 13,”Col NO : $”

PA equ 9800H

PB equ 9801H

PC equ 9802H

CR equ 9803H

mov ax,@data

mov ds,ax

mov al,90H

mov dx,CR

out dx,al

up1 : mov al,07H

mov dx,PC

out dx,al

mov dx,PA

in al,dx

cmp al,00H
```

```
je up1
call delay ; to avoid key bounce problem
lea si, array
mov bh,01H
mov al,01H
mov dx,PC
out dx,al
mov dx,PA
in al,dx
cmp al,00H
je row2
up2 : mov bl,01H
back : ror al,01H
jc disp
inc bl
inc si
cmp bl,08H
jbe back
disp : lea dx,m1
mov ah,09H
int 21H
add [si],30H
mov dl,[si]
mov ah,02H
int 21H
lea dx,m2
mov ah,09H
```

```
int 21H
add bh,30H
mov dl,bh
mov ah,02H
int 21H
lea dx,m3
mov ah,09H
int 21H
add bl,30H
mov dl,bl
mov ah,02H
int 21H
mov ah,4cH
int 21H
row2 : inc bh
add si,08H
mov al,02H
mov dx,PC
out dx,al
mov dx,PA
in al,dx
cmp al,00H
je row3
jmp up2
row3 : inc bh
add si,08H
mov al,04H
```

```
mov dx,PC
out dx,al
mov dx,PA
in al,dx
jmp up2
delay PROC
push bx
push cx
mov bx,0FFFFH
up3 : dec bx
jnz up3
pop cx
pop bx
RET
delay ENDP
end
```