

2.

a) Write two ALP modules stored in two different files; one module is to read a character from

the keyboard and the other one is to display a character. Use the above two modules to read a

string of characters from the keyboard terminated by the carriage return and print the string on

the display in the next line.

```
read1.asm (macro)
```

```
read macro
```

```
mov ah,01h
```

```
int 21h
```

```
endm
```

```
disp.asm (macro)
```

```
disp macro
```

```
mov ah,02h
```

```
int 21h
```

```
endm
```

```
.model small
```

```
.stack
```

```
.data
```

```
.code
```

```
up: read
```

```
str db 50 dup(0)
```

```
mov ax,@data
```

```
mov ds,ax
```

```
lea si,str
```

```
mov cx,00h
```

```
Include read1.asm
```

```
Include disp.asm
```

```
cmp al,0dh
```

```
je down
```

```
mov [si],al
```

```
inc si
```

```
inc cx
```

```
jmp up
```

```
down: mov dl,0ah
```

```
mov ah,02h
```

```
int 21h
```

```
up1: mov dl,[si]
```

```
disp
```

```
inc si
```

```
loop up1
```

```
mov ah,4ch
```

```
int 21h
```

b) Implement a BCD Up-Down Counter on the Logic Controller Interface.

```
.model small
```

```
.stack
```

```
.data
```

```
lea si,str
```

```
end
```

```
.code
```

```
PA equ 9800H
```

```
PB equ 9801H
```

```
PC equ 9802H
```

```
CR equ 9803H
mov ax,@data
mov ds,ax
mov al,80h
mov dx,CR
out dx,al
mov dx,PA
mov al,00h
up1: out dx,al
    call delay
    cmp al,99H
    je down1
    add al,01H
    DAA
    jmp up1
down1: out dx,al
    call delay
    sub al,01H
    DAS
    cmp al,00H
    jae down1
    mov ah,4cH
    int 21H
delay PROC
push ax
mov cx,0FFFFH
up3: mov bx,4FFFH
```

```
up4: dec bx
      jnz up4
      loop up3
      pop cx
      RET
delay ENDP
end
```