MC Codes

1) Write an embedded C program for interfacing button, LED, relay & buzzer as follows

a) When Switch 0 is pressed, LED's start chasing from left to right.

b) When Switch 1 is pressed, LED's start chasing from right to left.

```
#include <p18f4550.h>

#include "vector_relocate.h"

void delay(unsigned int time)

{

        unsigned int i,j;

        for(i=0;i<time;i++)

        {

                for(j=0;j<710;j++);

        }

}

void main(void)

{

        unsigned int i;

        unsigned char val=0;

        INTCON2bits.RBPU = 0;

        ADCON1=0x0F;

        TRISD = 0x00;

        TRISBbits.TRISB0=1;

        TRISBbits.TRISB1=1;

        while(1)

        {

                if(PORTBbits.RB1==0)

                {

                        val=1;

                }

                if(PORTBbits.RB0==0)

                {
```

```
            val=2;
        }
        if(val==1)
        {
            PORTD = 0x80;
            delay(100);
            for(i=0;i<8;i++)
            {
                PORTD=PORTD>>1;
                delay(100);
            }
        }
        if(val==2)
        {
            PORTD = 0x01;
            delay(100);
            for(i=0;i<8;i++)
            {
                PORTD=PORTD<<1;
                delay(100);
            }
        }
    }
}
```

2) Write an embedded C program for interfacing button, LED, relay & buzzer as follows

a) When Switch 0 is pressed, relay and buzzer is turned ON.

b) When Switch 1 is pressed, relay and buzzer is turned OFF & Flash LED’s.

```
#include <p18f4550.h>
#include "vector_relocate.h"
void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
    {
        for(j=0;j<710;j++);
    }
}
void main(void)
{
    unsigned char val=0;
    INTCON2bits.RBPU = 0;
    ADCON1=0x0F;
    TRISD = 0x00;
    TRISBbits.TRISB0=1;
    TRISBbits.TRISB1=1;
    TRISCbits.TRISC1=0;
    TRISCbits.TRISC2=0;
    PORTCbits.RC1=0;
    PORTCbits.RC2=0;
    while(1)
    {
        if(PORTBbits.RB1==0)
        {
            val=1;
        }
        if(PORTBbits.RB0==0)
        {
            val=2;
        }
        if(val==1)
        {
            PORTCbits.RC1=1;
            PORTCbits.RC2=1;
        }
        if(val==2)
        {
            PORTCbits.RC1=0;
            PORTCbits.RC2=0;
            PORTD = 0x55;
            delay(100);
            PORTD = 0xAA;
            delay(100);
        }
    }
}
```

3) Write an embedded C program for interfacing button, LED, relay & buzzer as follows

a) When Switch 0 is pressed relay and buzzer is turned ON and LED's start chasing from left to right.

b) When Switch 1 is pressed relay and buzzer is turned OFF and LED start chasing from right to left.

Code from Kale ma'am

```
#include <p18f4550.h>
#include "vector_relocate.h"
void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
    {
        for(j=0;j<710;j++);
    }
}
void main(void)
{
    unsigned int i;
    unsigned char val=0;
    INTCON2bits.RBPU = 0;
    ADCON1=0x0F;
```

```
TRISD = 0x00;
TRISBbits.TRISB0=1;
TRISBbits.TRISB1=1;
TRISCbits.TRISC1=0;
TRISCbits.TRISC2=0;
PORTCbits.RC1=0;
PORTCbits.RC2=0;
while(1)
{
        if(PORTBbits.RB1==0)
        {
                val=1;
        }
        if(PORTBbits.RB0==0)
        {
                val=2;
        }
        if(val==1)
        {
                PORTCbits.RC1=1;
                PORTCbits.RC2=1;
                PORTD = 0x80;
                delay(100);
                for(i=0;i<8;i++)
                {
                        PORTD=PORTD>>1;
                        delay(100);
                }
        }
        if(val==2)
        {
                PORTCbits.RC1=0;
```

```
            PORTCbits.RC2=0;
            PORTD = 0x01;
            delay(100);
            for(i=0;i<8;i++)
            {
                PORTD=PORTD<<1;
                delay(100);
            }
        }
    }
}
```

OR

Code from Turuk ma’am

```
//Includes
#include <p18f4550.h>                 //Include Controller specific .h
#include "vector_relocate.h"          //Vector Remapping for USB HID Bootloader
//Declarations
#define lrbit PORTBbits.RB1           //SW0 interfaced to RB1
#define rlbit PORTBbits.RB0           //SW1 interfaced to RB0
#define buzzer PORTCbits.RC2          //Buzzer interfaced to RC2
#define relay PORTDbits.RD7           //Relay interfaced to RC1

//Function Prototypes
void msdelay (unsigned int time);     //Function for delay
//Start of Program Code
void main()     //Main Program
{
    unsigned char i,val=0;        //Variable to latch the switch condition
    INTCON2bits.RBPU=0;           //To Activate the internal pull on PORTB
    ADCON1 = 0x0F;                //To disable the all analog inputs
```

```
TRISBbits.RB0=1;            //To configure RB0 as input for sensing SW1
TRISBbits.RB1=1;            //To configure RB1 as input for sensing SW0
TRISDbits.TRISD7=0;        //To configure RC1 (relay) as output
TRISCbits.TRISC2=0;       //To configure RC2 (buzzer) as output
TRISA = 0x00;               //To configure PORTD (LED) as output

PORTA = 0x00;            //Initial Value for LED
buzzer = 0;                //Initial Value for Buzzer
relay = 0;                  //Initial Value for Relay
while (1)                   //While loop for repeated operation
{
        if (lrbit==0)       //To check whether SW0 is pressed
        val = 1;            // Latch the status of switch SW0
        if (rlbit==0)       //To check whether SW1 is pressed
        val = 2;            // Latch the status of switch SW1

        if (val == 1)
        {
                buzzer = 1;
                relay = 1;
                // 6led
                // 0010 0000
                PORTA = 0x20;
                msdelay(50);
                for(i=0;i<6;i++)
                {
                        PORTA = PORTA >> 1;      //Shift right by 1 bit
                        msdelay(50);              // Make the MSB bit equal to 1
                }
        }
        if (val == 2)
        {
```

```
                        buzzer = 0;
                        relay = 0;
                        PORTA = 0x01;
                        // 0000 0001
                        msdelay(50);
                        for(i=0;i<6;i++)
                        {
                                PORTA = PORTA << 1;     //Shift left by 1 bit
                                msdelay(50);                 //Make the MSB bit equal to 1
                        }
                }
         }
}
//End of the Program
//Function Definitions
void msdelay (unsigned int time)          //Function for delay
{
        unsigned int i, j;
         for (i = 0; i < time; i++)
        {
                for (j = 0; j < 710; j++);         //Calibrated for a 1 ms delay in MPLAB
        }
}
```

4) Write an embedded C program for interfacing button, LED, relay & buzzer as follows

a) When Switch 0 is pressed relay and buzzer is turned ON and LED's start blinking.

b) When Switch 1 is pressed relay and buzzer is turned OFF and LED start chasing from left to right in a group of two.

```
#include <p18f4550.h>
#include "vector_relocate.h"
void delay(unsigned int time)
{
        unsigned int i,j;
        for(i=0;i<time;i++)
        {
                for(j=0;j<710;j++);
        }
}
void main(void)
{
        unsigned int i;
        unsigned char val=0;
        INTCON2bits.RBPU = 0;
        ADCON1=0x0F;
        TRISD = 0x00;
        TRISBbits.TRISB0=1;
        TRISBbits.TRISB1=1;
        TRISCbits.TRISC1=0;
        TRISCbits.TRISC2=0;
        PORTCbits.RC1=0;
        PORTCbits.RC2=0;
        while(1)
        {
                if(PORTBbits.RB1==0)
                {
                        val=1;
                }
                if(PORTBbits.RB0==0)
                {
                        val=2;
                }
                if(val==1)
                {
                        PORTCbits.RC1=1;
```

```
			PORTCbits.RC2=1;
			PORTD = 0xFF;
			delay(100);
			PORTD = 0x00;
			delay(100);
		}
		if(val==2)
		{
			PORTCbits.RC1=0;
			PORTCbits.RC2=0;
			PORTD = 0xC0;		//1100 0000
			delay(100);
			PORTD = 0x30;		//0011 0000
			delay(100);
			PORTD = 0x0C;		//0000 1100
			delay(100);
			PORTD = 0x03;		//0000 0011
			delay(100);
		}
	}
}
```

5) Write an embedded C program for interfacing button, LED, relay & buzzer as follows

a) When Switch 0 is pressed relay and buzzer is turned ON and LED's start blinking alternatively.

b) When Switch 1 is pressed relay and buzzer is turned OFF and LED start chasing from left to right in a group four.

```
#include <p18f4550.h>
#include "vector_relocate.h"
void delay(unsigned int time)
{
	unsigned int i,j;
	for(i=0;i<time;i++)
```

```
        {
                for(j=0;j<710;j++);
        }
}
void main(void)
{
        unsigned int i;
        unsigned char val=0;
        INTCON2bits.RBPU = 0;
        ADCON1=0x0F;
        TRISD = 0x00;
        TRISBbits.TRISB0=1;
        TRISBbits.TRISB1=1;
        TRISCbits.TRISC1=0;
        TRISCbits.TRISC2=0;
        PORTCbits.RC1=0;
        PORTCbits.RC2=0;
        while(1)
        {
                if(PORTBbits.RB1==0)
                {
                        val=1;
                }
                if(PORTBbits.RB0==0)
                {
                        val=2;
                }
                if(val==1)
                {
                        PORTCbits.RC1=1;
                        PORTCbits.RC2=1;
                        PORTD = 0xFF;
```

```
            delay(100);
            PORTD = 0x00;
            delay(100);
        }
        if(val==2)
        {
            PORTCbits.RC1=0;
            PORTCbits.RC2=0;
            PORTD = 0xF0;      //1111 0000
            delay(100);
            PORTD = 0x0F;     //0000 1111
            delay(100);
        }
    }
}
```

6) Write an embedded C program for interfacing an array of 8 LEDs to PIC18F4550.
Implement LED flashing.

```
#include <p18f4550.h>
#include "vector_relocate.h"
void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
    {
        for(j=0;j<710;j++);
    }
}
void main(void)
{
    ADCON1=0x0F;
    TRISD = 0x00;
```

```
        while(1)

        {

                PORTD = 0x55;

                delay(100);

                PORTD = 0xAA;

                delay(100);

        }

}
```

7) **Write an embedded C program** to generate square wave of 10Hz using timer0 with interrupt and blink LEDs after a delay of 50ms. Show detailed calculations.

Calculations:
Fosc=48MHz, N=16, F=10Hz
Tosc=(1/Fosc), T=(1/2F)=50ms
x = T / (4*N*Tosc)
= 0.05 / 1.33u (u=10^-6)
x = 37594
Count value = 65535 + 1 – x = 27942
Hex value of 27942 = 6D26

```
        #include <p18f4550.h>

        extern void _startup (void);

        void timer_isr(void);

        #pragma code RESET_INTERRUPT_VECTOR = 0x1000

        void _reset (void)

        {

                _asm

                        goto _startup

                _endasm

        }

        #pragma code

        void main(void)

        {

                ADCON1 = 0x0F;

                INTCON2bits.RBPU=0;

                TRISBbits.TRISB0 = 0;
                TRISD = 0x00;
                PORTBbits.RB0 = 0;
                PORTD = 0x00;
                T0CON = 0x03;
                TMR0H = 0x6D;
                TMR0L = 0x26;
                INTCONbits.TMR0IF = 0;
                INTCONbits.TMR0IE = 1;
                INTCONbits.GIE = 1;
                T0CONbits.TMR0ON = 1;
                while(1);
        }
        #pragma code HIGH_INTERRUPT_VECTOR = 0x1008
        void high_ISR (void)
        {
                _asm
                        goto timer_isr
                _endasm
        }
        #pragma code
        ///Timer ISR Function Prototype//
        #pragma interrupt timer_isr
        void timer_isr(void)
        {
                TMR0H = 0X6D;
                TMR0L = 0X26;
                PORTBbits.RB0 = ~PORTBbits.RB0;
                PORTD = ~PORTD;
                INTCONbits.TMR0IF = 0;
        }
```

8) **Write an embedded C program** to generate square wave of 100 Hz using timer0 with interrupt. Show detailed calculations.

Calculations:
Fosc=48MHz, N=16, F=100Hz
Tosc=(1/Fosc), T=(1/2F)=5ms
x = T / (4*N*Tosc)
= 0.005 / 1.33u (u=10^-6)
x = 3759
Count value = 65535 + 1 – x = 61777
Hex value of 61777 = F151

```
#include <p18f4550.h>
extern void _startup (void);
void timer_isr(void);
#pragma code RESET_INTERRUPT_VECTOR = 0x1000
void _reset (void)
{
        _asm
              goto _startup
        _endasm
}
#pragma code
void main(void)
{
        ADCON1 = 0x0F;
        INTCON2bits.RBPU=0;
        TRISBbits.TRISB0 = 0;
        PORTBbits.RB0 = 0;
        T0CON = 0x03;
        TMR0H = 0xF1;
        TMR0L = 0x51;
        INTCONbits.TMR0IF = 0;
        INTCONbits.TMR0IE = 1;
        INTCONbits.GIE = 1;
        T0CONbits.TMR0ON = 1;
        while(1);
}
#pragma code HIGH_INTERRUPT_VECTOR = 0x1008
```

```
void high_ISR (void)
{
        _asm
                goto timer_isr
        _endasm
}
#pragma code
///Timer ISR Function Prototype//
#pragma interrupt timer_isr
void timer_isr(void)
{
        TMR0H = 0XF1;
        TMR0L = 0X51;
        PORTBbits.RB0 = ~PORTBbits.RB0;
        INTCONbits.TMR0IF = 0;
}
```

9) **Write an embedded C program** to control the speed of DC motor using PWM. Assume the PWM frequency as 4 KHz with duty cycle of 20% and 40%.

Calculation:
PR2=187
CCPR1L = 20% of 187
= 37.4
Hex value of 37 = 25

CCPR1L = 40% of 187
= 74.8
Hex value of 74 = 4A

DC1B1	**DC1B0**	**Decimal value**
0	**0**	**0**
0	**1**	**0.25**
1	**0**	**0.5**
1	**1**	**0.75**

```
#include <p18f4550.h>
#include "vector_relocate.h"
void myMsDelay(unsigned int time);
```

```
}


void main() {
    TRISCbits.TRISC2 = 0;
    TRISD = 0;
    PR2 = 187;
    CCP1CON = 0x0C;
    T2CON = 0x07;
    PORTDbits.RD5 = 1;
    PORTDbits.RD6 = 0;

    while (1) {
        // For 20% Duty cycle
        CCPR1L = 0x25;
        CCP1CONbits.DC1B0 = 0;
        CCP1CONbits.DC1B1 = 1;
        myMsDelay(3000);

        // For 40% Duty cycle
        CCPR1L = 0x4A;
        CCP1CONbits.DC1B0 = 1;
        CCP1CONbits.DC1B1 = 1;
        myMsDelay(3000);
    }
}

void myMsDelay(unsigned int time)
{
    unsigned int i, j;
    for (i = 0; i < time; i++)
    {
        for (j = 0; j < 710; j++);
    }
```

```
}
```

10) Write an embedded C program to control the speed of DC motor using PWM. Assume the PWM frequency as 4 KHz with duty cycle of 60% and 80%.

Calculation:
PR2=187
CCPR1L = 60% of 187
= 112.2
Hex value of 112 = 70

CCPR1L = 80% of 187
= 149.6
Hex value of 149 = 95

DC1B1	**DC1B0**	**Decimal value**
0	**0**	**0**
0	**1**	**0.25**
1	**0**	**0.5**
1	**1**	**0.75**

```
#include <p18f4550.h>

#include "vector_relocate.h"

void myMsDelay(unsigned int time);


void main() {

   TRISCbits.TRISC2 = 0;

   TRISD = 0;

   PR2 = 187;

   CCP1CON = 0x0C;

   T2CON = 0x07;

   PORTDbits.RD5 = 1;

   PORTDbits.RD6 = 0;


   while (1) {

      // For 60% Duty cycle

      CCPR1L = 0x70;

      CCP1CONbits.DC1B0 = 1;

      CCP1CONbits.DC1B1 = 0;
```

```
        myMsDelay(3000);


        // For 80% Duty cycle
        CCPR1L = 0x95;
        CCP1CONbits.DC1B0 = 0;
        CCP1CONbits.DC1B1 = 1;
        myMsDelay(3000);
    }
}


void myMsDelay(unsigned int time)
{
    unsigned int i, j;
    for (i = 0; i < time; i++)
    {
        for (j = 0; j < 710; j++);
    }
}
```

11) **Write an embedded C program** to control the speed of DC motor using PWM. Assume the PWM frequency as 4 KHz with duty cycle of 50%. Motor should rotate in clockwise and anticlockwise direction.

Calculation:
PR2=187
CCPR1L = 50% of 187
= 93.5
Hex value of 93 = 5D

DC1B1	**DC1B0**	**Decimal value**
0	**0**	**0**
0	**1**	**0.25**
1	**0**	**0.5**
1	**1**	**0.75**

```
#include <p18f4550.h>
#include "vector_relocate.h"
void myMsDelay(unsigned int time);
void main() {
    TRISCbits.TRISC2 = 0;
    TRISD = 0;
    PR2 = 187;
    CCP1CON = 0x0C;
    T2CON = 0x07;
    PORTDbits.RD5 = 1;
    PORTDbits.RD6 = 0;

    while (1)
    {
        // For 50% Duty cycle
        CCPR1L = 0x5D;
        CCP1CONbits.DC1B0 = 0;
        CCP1CONbits.DC1B1 = 1;
        myMsDelay(3000);
    }
}

void myMsDelay(unsigned int time)
{
    unsigned int i, j;
    for (i = 0; i < time; i++)
    {
        for (j = 0; j < 710; j++);
    }
}
```

12) Write an embedded C program

Interface LCD (8 bit mode) to **PIC18F4550** to display a message “Welcome” on 1st line and “To Embedded Lab” on 2nd line of LCD. Draw a neat interfacing diagram for the same.

```
#include<p18f4550.h>
```

```
#include "vector_relocate.h"
#define LCD_DATA PORTD
#define ctrl PORTE
#define rs PORTEbits.RE0
#define rw PORTEbits.RE1
#define en PORTEbits.RE2
void init_LCD(void);
void LCD_command(unsigned char cmd);
void LCD_data(unsigned char data);
void LCD_write_string(static char *str);
void msdelay (unsigned int time);

void main( void )
{
    char var1[] = "Welcome";
    char var2[ ] = "To Embedded Lab";
    INTCON2bits.RBPU=0;
    ADCON1=0x0F;
    TRISD=0x00;
    TRISE=0x00;
    init_LCD();
    msdelay(15);
    LCD_write_string(var1);
    msdelay(15);
    LCD_command(0xC0);
    LCD_write_string(var2);
    while(1);
}

void init_LCD(void)
{
    LCD_command(0x38);
```

```
        msdelay(15);

        LCD_command(0x01);

        msdelay(15);

        LCD_command(0x0E);

        msdelay(15);

        LCD_command(0x80);

        msdelay(15);

}


void LCD_command(unsigned char cmd)

{

        LCD_DATA = cmd;

        rs = 0;

        rw = 0;

        en = 1;

        msdelay(15);

        en = 0;

}


void LCD_data(unsigned char data)

{

        LCD_DATA = data;

        rs = 1;

        rw = 0;

        en = 1;

        msdelay(15);

        en = 0;

}


void LCD_write_string(static char *str)

{

        int i = 0;
```

```
    while (str[i] != '\0')

    {

        LCD_data(str[i]);

        msdelay(15);

        i++;

    }

}


void msdelay(unsigned int t)

{

    unsigned int i,j;

    for(i=0;i<t;i++)

    {

        for(j=0;j<710;j++);

    }

}
```

13) Write an embedded C program

Interface LCD (8 bit mode) to **PIC18F4550** to display a message “Name of Student” on 1st line and “Seat No.” on 2nd line of LCD. Draw a neat interfacing diagram for the same.

```
#include<p18f4550.h>

#include "vector_relocate.h"

#define LCD_DATA PORTD

#define ctrl PORTE

#define rs PORTEbits.RE0

#define rw PORTEbits.RE1

#define en PORTEbits.RE2

void init_LCD(void);

void LCD_command(unsigned char cmd);

void LCD_data(unsigned char data);

void LCD_write_string(static char *str);

void msdelay (unsigned int time);
void main( void )
{
    char var1[] = "Anurag Thorat";        //Your name
    char var2[ ] = "T1900503321";       //Your seat number
    INTCON2bits.RBPU=0;
    ADCON1=0x0F;
    TRISD=0x00;
    TRISE=0x00;
    init_LCD();
    msdelay(15);
    LCD_write_string(var1);
    msdelay(15);
    LCD_command(0xC0);
    LCD_write_string(var2);
    while(1);
}

void init_LCD(void)
{
    LCD_command(0x38);
    msdelay(15);
    LCD_command(0x01);
    msdelay(15);
    LCD_command(0x0E);
    msdelay(15);
    LCD_command(0x80);
    msdelay(15);
}

void LCD_command(unsigned char cmd)
{
    LCD_DATA = cmd;
    rs = 0;
    rw = 0;
    en = 1;
    msdelay(15);
    en = 0;
}

void LCD_data(unsigned char data)
{
    LCD_DATA = data;
    rs = 1;
    rw = 0;
    en = 1;
    msdelay(15);
    en = 0;
}

void LCD_write_string(static char *str)
{
    int i = 0;
    while (str[i] != '\0')
    {
            LCD_data(str[i]);
            msdelay(15);
            i++;
    }
}

void msdelay(unsigned int t)
{
    unsigned int i,j;
    for(i=0;i<t;i++)
    {
            for(j=0;j<710;j++);
    }
}
```

14) Write an embedded C program

Interface LCD (8 bit mode) to **PIC18F4550** to display a message “Microcontroller” on 1st line and “Embedded Lab” on 2nd line of LCD. Draw a neat interfacing diagram for the same.

```
#include<p18f4550.h>
#include "vector_relocate.h"
#define LCD_DATA PORTD
#define ctrl PORTE
#define rs PORTEbits.RE0
#define rw PORTEbits.RE1
#define en PORTEbits.RE2
void init_LCD(void);
void LCD_command(unsigned char cmd);
void LCD_data(unsigned char data);
void LCD_write_string(static char *str);
void msdelay (unsigned int time);


void main( void )
{
    char var1[] = "Microcontroller";
    char var2[ ] = "Embedded Lab";
    INTCON2bits.RBPU=0;
    ADCON1=0x0F;
    TRISD=0x00;
    TRISE=0x00;
    init_LCD();
    msdelay(15);
    LCD_write_string(var1);
    msdelay(15);
    LCD_command(0xC0);
    LCD_write_string(var2);
    while(1);
}

void init_LCD(void)
{
    LCD_command(0x38);
    msdelay(15);
    LCD_command(0x01);
    msdelay(15);
    LCD_command(0x0E);
    msdelay(15);
    LCD_command(0x80);
    msdelay(15);
}

void LCD_command(unsigned char cmd)
{
    LCD_DATA = cmd;
    rs = 0;
    rw = 0;
    en = 1;
    msdelay(15);
    en = 0;
}

void LCD_data(unsigned char data)
{
    LCD_DATA = data;
    rs = 1;
    rw = 0;
    en = 1;
    msdelay(15);
    en = 0;
}

void LCD_write_string(static char *str)
{
    int i = 0;
    while (str[i] != '\0')
    {
            LCD_data(str[i]);
            msdelay(15);
            i++;
    }
}

void msdelay(unsigned int t)
{
    unsigned int i,j;
    for(i=0;i<t;i++)
    {
            for(j=0;j<710;j++);
    }
}
```

15) **Write an embedded C program** to interface analog voltage 0-5V to internal ADC and display value on LCD.

```
#include <p18f4550.h>
#include "vector_relocate.h"
#define LCD_DATA  PORTD             // LCD data port
#define en PORTEbits.RE2      // enable signal
#define rw PORTEbits.RE1      // read/write signal
```

```
#define rs PORTEbits.RE0       // register select signal

void LCD_cmd(unsigned char cmd);
void myMsDelay(unsigned int time) {
    unsigned int i, j;
    for (i = 0; i < time; i++)
   {
        for (j = 0; j <710; j++);
   }
}

void init_LCD(void) {
    LCD_cmd(0x38);   // initialization of 16X2 LCD in 8bit mode
    myMsDelay(15);

    LCD_cmd(0x01);   // clear LCD
    myMsDelay(15);

    LCD_cmd(0x0E);   // cursor off
    myMsDelay(15);

    LCD_cmd(0x80);
    myMsDelay(15);
}

// Function to pass command to the LCD
void LCD_cmd(unsigned char cmd) {
    LCD_DATA = cmd;
    rs = 0;
    rw = 0;
    en = 1;
    myMsDelay(15);
    en = 0;
    myMsDelay(15);
}

// Function to write data to the LCD
void LCD_write(unsigned char data) {
    LCD_DATA = data;
    rs = 1;
    rw = 0;
    en = 1;
    myMsDelay(15);
    en = 0;
    myMsDelay(15);
}

void main(void) {
    unsigned int val[4], ADC_Result = 0, var;
    unsigned char i, str[] = "Result:";

    TRISD = 0x00;    // Configuring PORTD as output
    TRISE = 0;
    TRISA = 0xFF;
    init_LCD();

    // ADC Initialization
    ADCON1 = 0x0C;
    ADCON2 = 0x8E;
    ADCON0 = 0x09;  // Turn ON ADC module

    LCD_cmd(0x80);
    for (i = 0; str[i] != '\0'; i++) {
        LCD_write(str[i]);
        myMsDelay(200);
    }

    while (1) {
        ADCON0bits.GO = 1;
        while (ADCON0bits.GO == 1);
        var = ((unsigned int)ADRESH) << 8;
        ADC_Result = var + ADRESL;

        for (i = 0; i < 4; i++) {
            val[i] = ADC_Result % 0x0A;
            val[i] = val[i] + 0x30;
            ADC_Result = ADC_Result / 0x0A;
        }

        LCD_cmd(0x87);     // LCD command to set DDRAM address
        LCD_write(val[3]);
        LCD_write(val[2]);
        LCD_write(val[1]);
        LCD_write(val[0]);
    }
}
```