1.Addition of two 8 bit nos.

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

void main(void)
{
        int  sum;
        sum =0;
        sum= 0x05+0X02;
        TRISD=0;
        PORTD=sum;

}
```

Program: addition of array of numbers

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f4550.h>
void main(void)
{

  int i,sum,n;
  int number[] = {1,2,3,4,5,6,7,8,9,10};    // array of 10 numbers
  sum = 0;                          // initialize sum as zero
    for(i=0; i<=9;i++)
        {                  //indexing start from 0 to 9
     sum = sum+number[i];
        }

  TRISB =0;                         //initialize Port_B as output
  PORTB = sum;                        // from sum to PORT_B

  //n = 0xFF + 0XFF;
}
```

a)Internal to internal memory transfer

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f4550.h>
/*
 *
 */
void main(void)
{

   int i;
   int source1[] = {0x21,0x22,0x23,0x24,0x25};   // source mem block
   int dest[] = {0x00,0x00,0x00,0x00,0x00};      // destination mem block

   for(i=0; i<=4;i++)
          {                    // counter = 5

     dest[i] = source1[i];                 // source to destination

          }
}
```

b)memory exchange program

```
#include <stdio.h>

#include <stdlib.h>

#include <pic18f4550.h>

void main(void)

{


  int temp,i;

  int source1[] = {0x21,0x22,0x23,0x24,0x25};

  int dest[] = {0x99,0x99,0x99,0x99,0x99};


  for(i=0; i<=4;i++)

        {

   temp = source1[i];

   source1[i] = dest[i];

   dest[i] = temp;


        }

}
```

Program to mul and div 8 bit no. by 8 bit

4. A multiplication

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f452.h>//multiplication using successive addition
void main(void) {
  //static int v_mem[] = 0x55;      @0x0005
  int num1, num2;
  int result,i;

  result = 0;
  num1 = 0x23;
  num2 = 0x10;

  for(i=1; i<=num2; i++)
  {
    result = result + num1;
  }
  TRISB =0;
  PORTB = result;
    TRISC=0;
  PORTC=num1*num2;

}
```

4 B Division

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f452.h>//Division using successive sustraction
void main(void) {
  //static int v_mem[] = 0x55;      @0x0005
  int dividend, divisor,quotient;
  int result,i;
  dividend = 0x0E;
   divisor= 0x04;
  result = dividend;
  quotient=0;
  while(1)
  {
    if(divisor==0)
      result=0;
    else if(result<divisor)
      break;
    else
    {
      result = result - divisor;
      quotient=quotient+1;

    }
  }
```

```
result = dividend;
quotient=0;
while(1)
{
  if(divisor==0)
    result=0;
  else if(result<divisor)
    break;
  else
  {
    result = result - divisor;
    quotient=quotient+1;

  }
}
```

```
TRISA=0;
PORTA=result; // remainder
TRISB =0;
PORTB = quotient;   //quotient

TRISC=0;
PORTC=dividend/divisor;

}
```

```
    TRISA=0;
    PORTA=result; // remainder
    TRISB =0;
    PORTB = quotient;   //quotient

    TRISC=0;
    PORTC=dividend/divisor;

}
```

Progrm for sorting of no. in asc and des order

5A.ASCENDING

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f4550.h>
void main(void)
 {
   int i,j,temp;
   int num_asc[] = {10,2,5,1,6};

   for(i=0; i<=4; i++)
        {              // point to LHS number
      for(j=i+1;j<=4;j++)            // point to RHS number
      if (num_asc[i] > num_asc[j])
                {    // if LHS > RHS , change the position
         temp = num_asc[i];
         num_asc[i] =num_asc[j];
         num_asc[j]= temp;
                }
       }
```

5 B descending

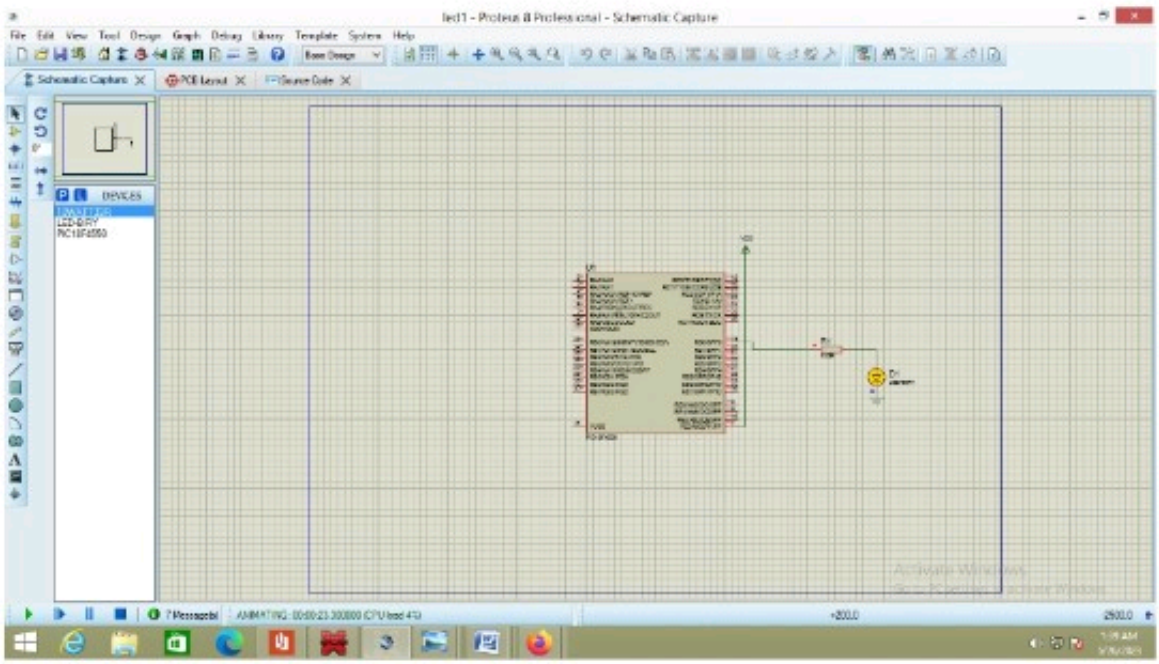
```
//        Sorting the array

#include <stdio.h>

#include <stdlib.h>

#include <pic18f452.h>

void main(void) {

   int i,j,temp;

   int num_asc[] = {7,2,5,1,6};

 for(i=0; i<=4; i++){              // point to LHS number

      for(j=i+1;j<=4;j++)           // point to RHS number

          // MSDELAY(55250);
```

```
        if (num_asc[i] < num_asc[j]){     // if LHS > RHS , change the position
        temp = num_asc[i];
        num_asc[i] =num_asc[j];
        num_asc[j]= temp;
      }
    }
    TRISA=0;
    for(i=0; i<=4; i++)
    {
      // MSDELAY(250);
      PORTA=num_asc[i];
    }
}
```

Program for LED interfacing:

```
#include<p18f4520.h>
void delay(unsigned int itime);
#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF
void main()
{
  TRISD=0;
  {
    while(1)
    {
      PORTD=0X00;
      delay(100);
      PORTD=0xff;
      delay(100);
    }
  }
}
void  delay(unsigned int itime)
{
  int i,j;
  for(i=0;i<=itime;i++)
    for(j=0;j<=1275;j++);
}
```

OUTPUT:


led1 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
PCB Layout
Source Code
DEVICES
LED-BIRY
PIC16F4550

Input:

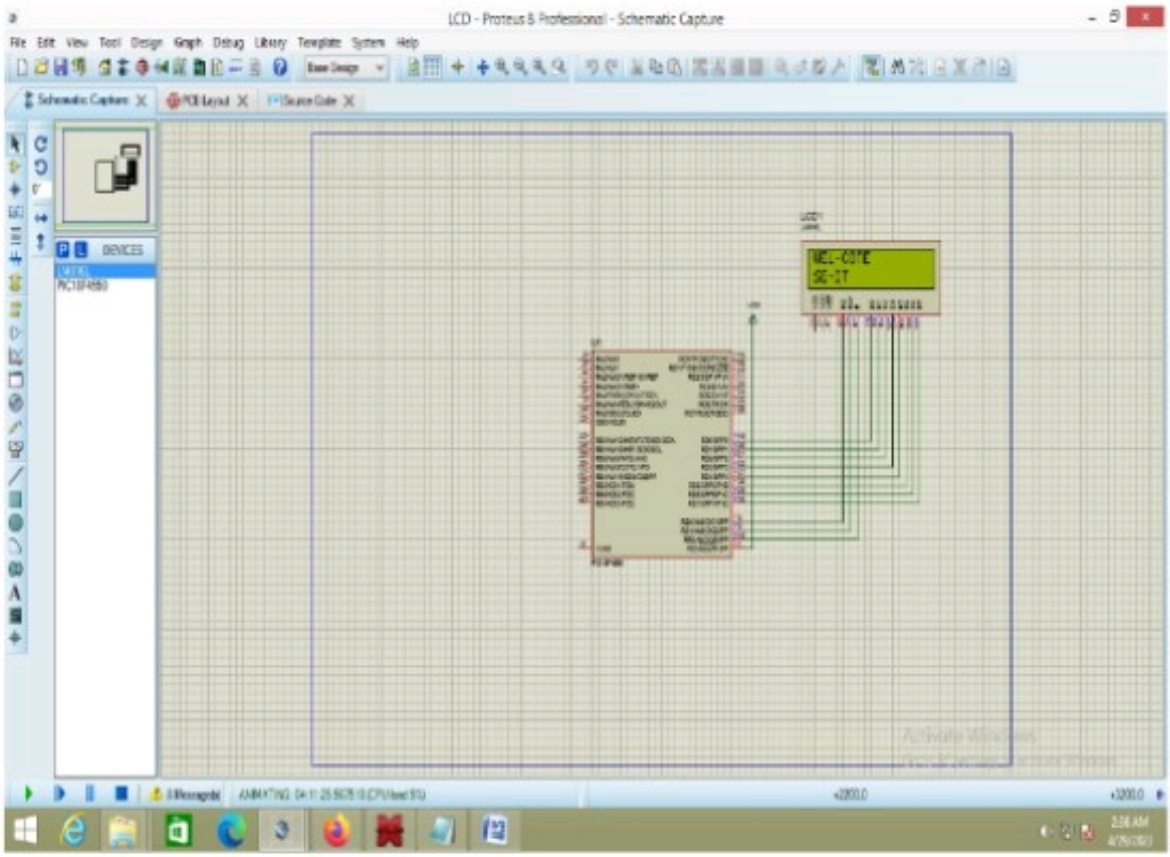
```
#include <PIC18f4550.h> //Include Controller specific .h
//#pragma config FOSC = HS //Oscillator Selection
#pragma config WDT = OFF //Disable Watchdog timer
#pragma config LVP = OFF //Disable Low Voltage Programming
//#pragma config PBADEN = OFF //Disable PORTB Analog inputs
//Declarations
#define LCD_DATA PORTD //LCD data port to PORTD
#define ctrl PORTE //LCD control port to PORTE
#define rs PORTEbits.RE0 //register select signal to RE0
#define rw PORTEbits.RE1 //read/write signal to RE1
#define en PORTEbits.RE2 //enable signal to RE2
//Function Prototypes
void init_LCD(void); //Function to initialize the LCD
void LCD_command(unsigned char cmd);//Function to pass command to LCD
void LCD_data(unsigned char data); //Function to write char to LCD
void LCD_write_string(char *str);//Function to write string
void msdelay (unsigned int time); //Function to generate delay
//Start of Main Program
void main(void)
{
char var1[] = "WEL-COME";//Declare message to be displayed
char var2[] = "SE-IT  ";
ADCON1 = 0x0F; //Configuring the PORTE pins as digital I/O
TRISD = 0x00; //Configuring PORTD as output
```

```
TRISE = 0x00; //Configuring PORTE as output
init_LCD(); // call function to initialize of LCD
msdelay(50); // delay of 50 milliseconds
LCD_write_string(var1); //Display message on first line
msdelay(150);
LCD_command(0xC0); // initiate cursor to second line
LCD_write_string(var2);//Display message on second line
while (1); //Loop here
} //End of Main

void msdelay (unsigned int time) //Function to generate 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
}
void init_LCD(void) // Function to initialize the LCD
{
LCD_command(0x38); // initialization of 16X2 LCD in 8bit mode
msdelay(15);
LCD_command(0x01); // clear LCD
msdelay(15);
LCD_command(0x0C); // cursor off
msdelay(15);
LCD_command(0x80); // go to first line and 0th position
```

```
msdelay(15);
}
void LCD_command(unsigned char cmd) //Function to pass command to LCD
{
LCD_DATA = cmd; //Send data on LCD data bus
rs = 0; //RS = 0 since command to LCD
rw = 0; //RW = 0 since writing to LCD
en = 1; //Generate High to low pulse on EN
msdelay(15);
en = 0;
}
void LCD_data(unsigned char data)//Function to write data to the LCD
{
LCD_DATA = data; //Send data on LCD data bus
rs = 1; //RS = 1 since data to LCD
rw = 0; //RW = 0 since writing to LCD
en = 1; //Generate High to low pulse on EN
msdelay(15);
en = 0;
}
//Function to write string to LCD
void LCD_write_string( char *str)
{
int i = 0;
while (str[i] != '\0') //Check for end of the string
```

```
{
LCD_data(str[i]); // sending data on LCD byte by byte
msdelay(15);
i++;
}
}
```

Output:

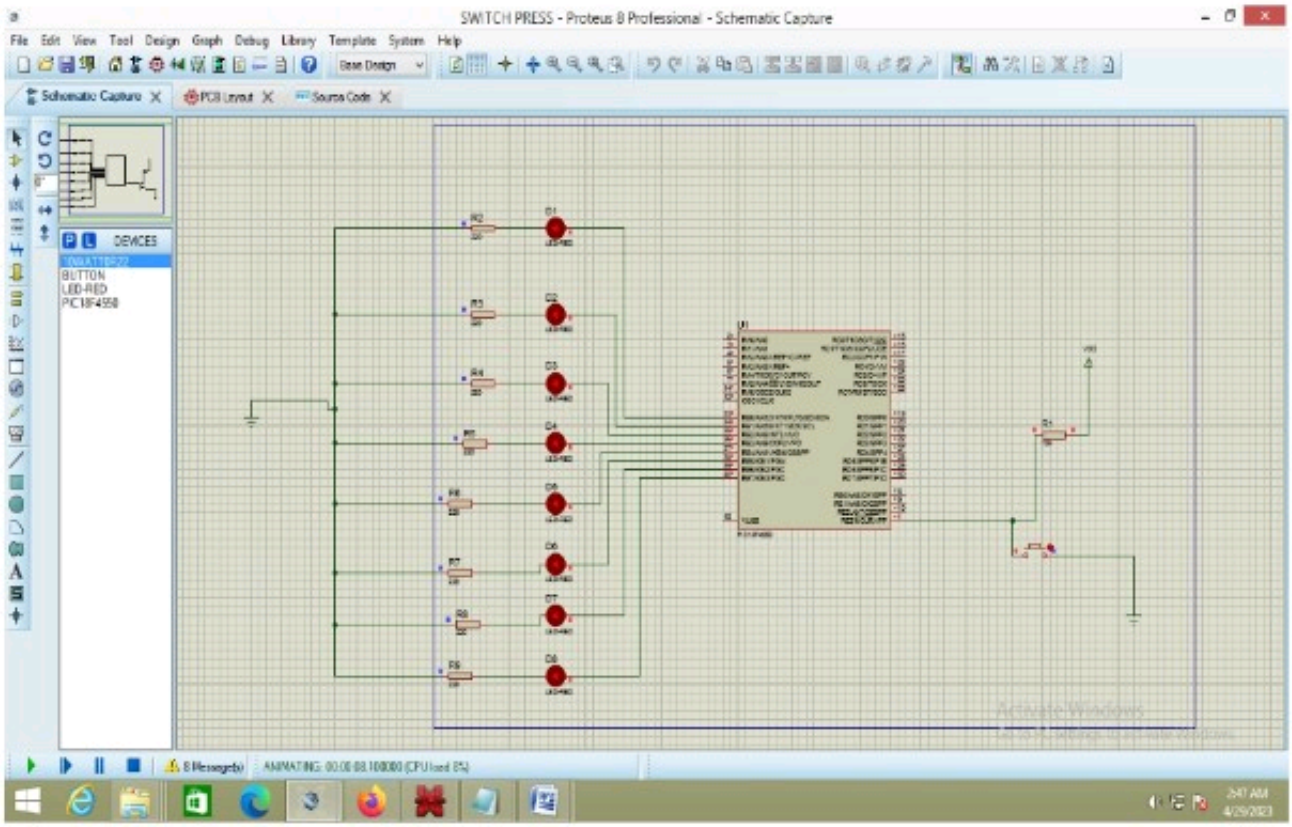


Input:

```
#include<xc.h>
#include<plib/delays.h>
#include <pic18f4550.h>
#define direction TRISB
void delay_ms(unsigned int val);
void main()
{
   OSCCON=0x72;
   direction=0;
    PORTB = 0;
   while(1)
   {
      PORTB = 0xFF;
      delay_ms(1000);
      PORTB = 0x00;
      delay_ms(1000);
   }
}
void delay_ms(unsigned int val)
{
   unsigned int i,j;
   for(i=0;i<val;i++)
      for(j=0;j<165;j++);
```

}

Output:

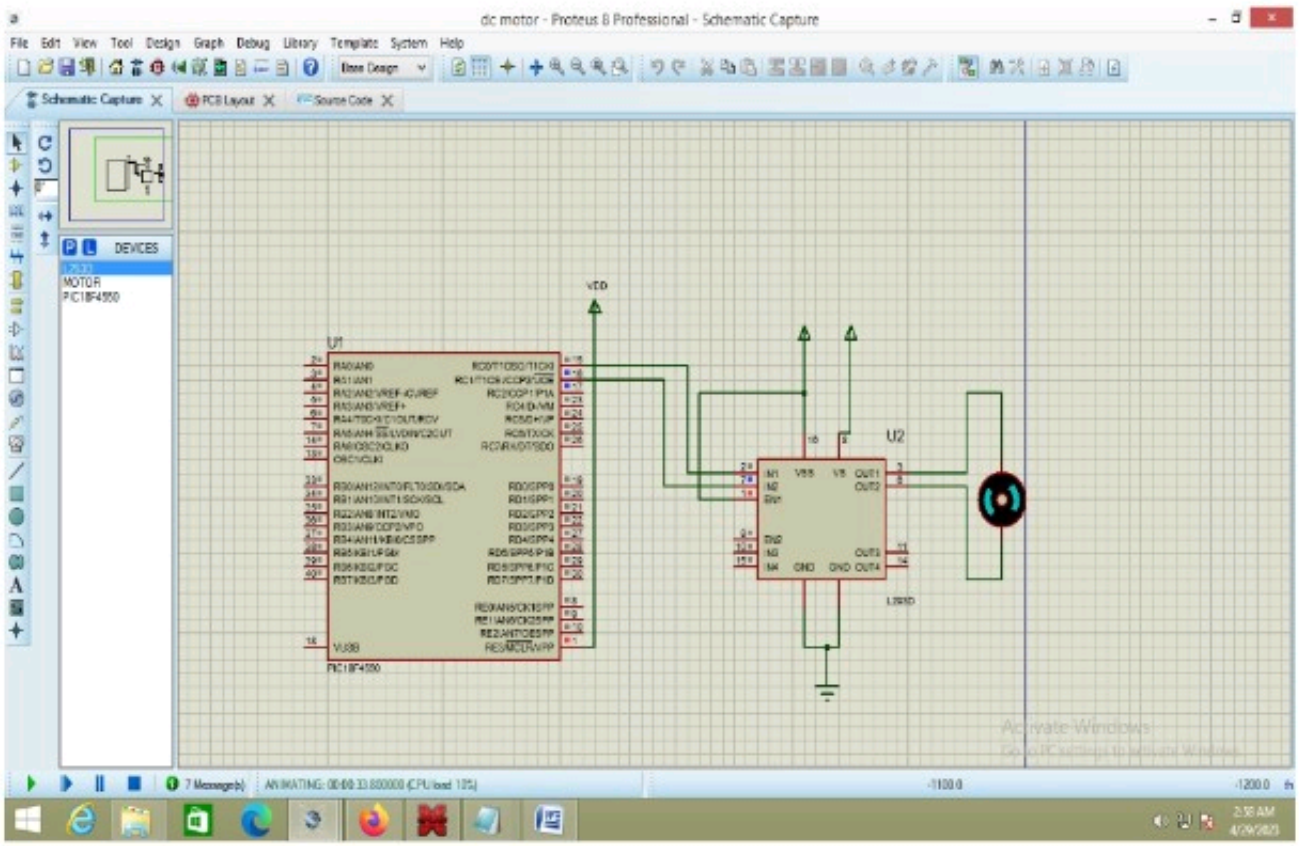

SWITCH PRESS - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
PCB Layout
Source Code
DEVICES
BUTTON
LED-RED
PIC18F4550

Input:

```
/*     Calculations
 * Fosc = 48MHz
 *
 * PWM Period = [(PR2) + 1] * 4 * TMR2 Prescale Value / Fosc
 * PWM Period = 200us
 * TMR2 Prescale = 16
 * Hence, PR2 =  149 or 0x95
 *
 * Duty Cycle = 10% of 200us
 * Duty Cycle = 20us
 * Duty Cycle = (CCPR1L:CCP1CON<5:4>) * TMR2 Prescale Value / Fosc
 * CCP1CON<5:4> = <1:1>
 * Hence, CCPR1L = 15 or 0x0F
 */


#include<p18f4550.h>

unsigned char count=0;
bit TIMER,SPEED_UP;

void timer2Init(void)
{
   T2CON   =   0b00000010;          //Prescalar = 16; Timer2 OFF
   PR2     =   0x95;               //Period Register
}


void delay(unsigned int time)
{
   unsigned int i,j;
   for(i=0;i<time;i++)
      for(j=0;j<1000;j++);
}


void main(void)
{
   unsigned int i;
   TRISCbits.TRISC1    = 0;         //RC1 pin as output
   TRISCbits.TRISC2    = 0;         //CCP1 pin as output
   LATCbits.LATC1      = 0;
   CCP1CON  =   0b00111100;         //Select PWM mode; Duty cycle LSB CCP1CON<4:5>
= <1:1>
   CCPR1L   =   0x0F;              //Duty cycle 10%
```

```
  timer2Init();                 //Initialise Timer2
  TMR2ON = 1;                   //Timer2 ON

  while(1)                      //Loop forever
  {
    for(i=15;i<150;i++)
    {
      CCPR1L = i;
      delay(100);
    }
    for(i=150;i>15;i--)
    {
      CCPR1L = i;
      delay(100);
    }
  }
}
```

Output:



```
#include <pic18f4550.h>
#include <stdio.h>

#define LCD_EN LATAbits.LA1
#define LCD_RS LATAbits.LA0
#define LCDPORT LATB

unsigned char str[16];

void lcd_delay(unsigned int time)
{
 unsigned int i , j ;

   for(i = 0; i < time; i++)
   {
     for(j=0;j<100;j++);
   }
}


void SendInstruction(unsigned char command)
{
   LCD_RS = 0;            // RS low : Instruction
   LCDPORT = command;
   LCD_EN = 1;            // EN High
   lcd_delay(10);
   LCD_EN = 0;            // EN Low; command sampled at EN falling edge
   lcd_delay(10);
}

void SendData(unsigned char lcddata)
{
   LCD_RS = 1;            // RS HIGH : DATA
   LCDPORT = lcddata;
   LCD_EN = 1;            // EN High
   lcd_delay(10);
   LCD_EN = 0;            // EN Low; data sampled at EN falling edge
   lcd_delay(10);
}

void InitLCD(void)
```

```
{
  ADCON1 = 0x0F;
  TRISB = 0x00; //set data port as output
  TRISAbits.RA0 = 0; //RS pin
  TRISAbits.RA1 = 0; // EN pin

  SendInstruction(0x38);    //8 bit mode, 2 line,5x7 dots
  SendInstruction(0x06);        //entry mode
  SendInstruction(0x0C);        //Display ON cursor OFF
  SendInstruction(0x01);    //Clear display
  SendInstruction(0x80);    //set address to 0
}

void LCD_display(unsigned int row, unsigned int pos, unsigned char *ch)
{
  if(row==1)
    SendInstruction(0x80 | (pos-1));
  else
    SendInstruction(0xC0 | (pos-1));

  while(*ch)
    SendData(*ch++);
}

void ADCInit(void)
{
  TRISEbits.RE2 = 1;            //ADC channel 7 input

  ADCON1 = 0b00000111;          //Ref voltages Vdd & Vss; AN0 - AN7 channels Analog
  ADCON2 = 0b10101110;          //Right justified; Acquisition time 4T; Conversion clock Fosc/64
}

unsigned short Read_Temp(void)
{
  ADCON0 = 0b00011101;    //ADC on; Select channel;
  GODONE = 1;           //Start Conversion

  while(GO_DONE == 1 );  //Wait till A/D conversion is complete
  return ADRES;                 //Return ADC result
}
```

```
int main(void)
{
   unsigned int temp;
   InitLCD();
   ADCInit();
   LCD_display(1,1,"Temperature:");
   while(1)
   {
      temp = Read_Temp();
      temp = ((temp * 500) / 1023);
      sprintf(str,"%d'C  ",temp);
      LCD_display(2,1,str);
      lcd_delay(9000);
   }
   return 0;
}
```