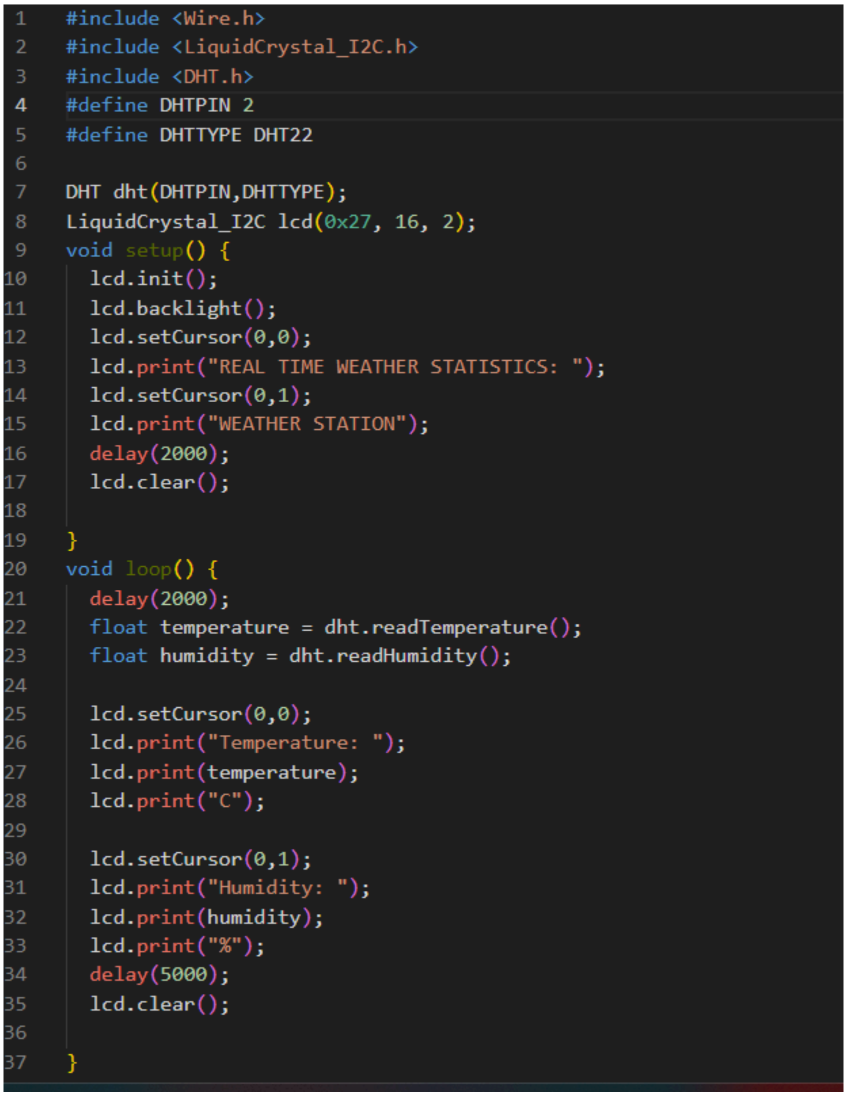
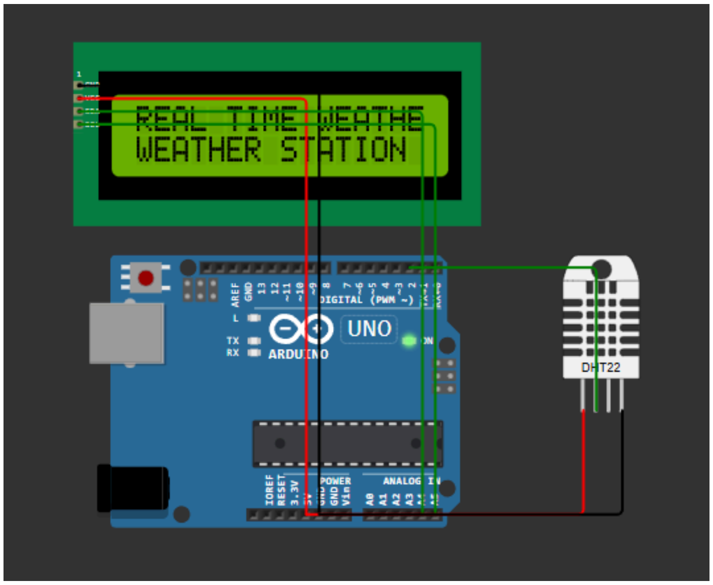
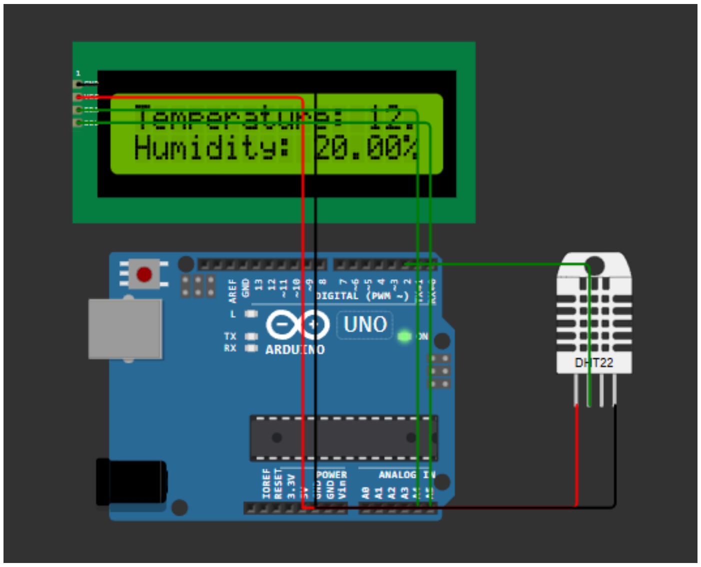
IOT HOLIDAY ASSIGNMENT

1. **Write an embedded C program to create a weather-reporting system that provides real-time environmental data to users**

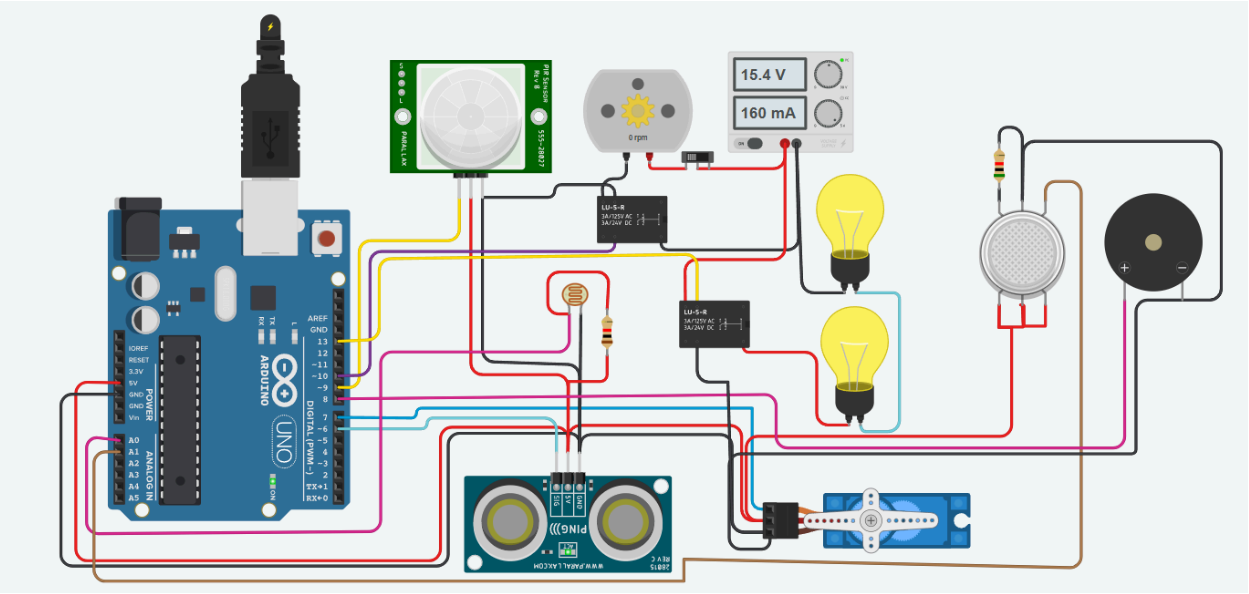


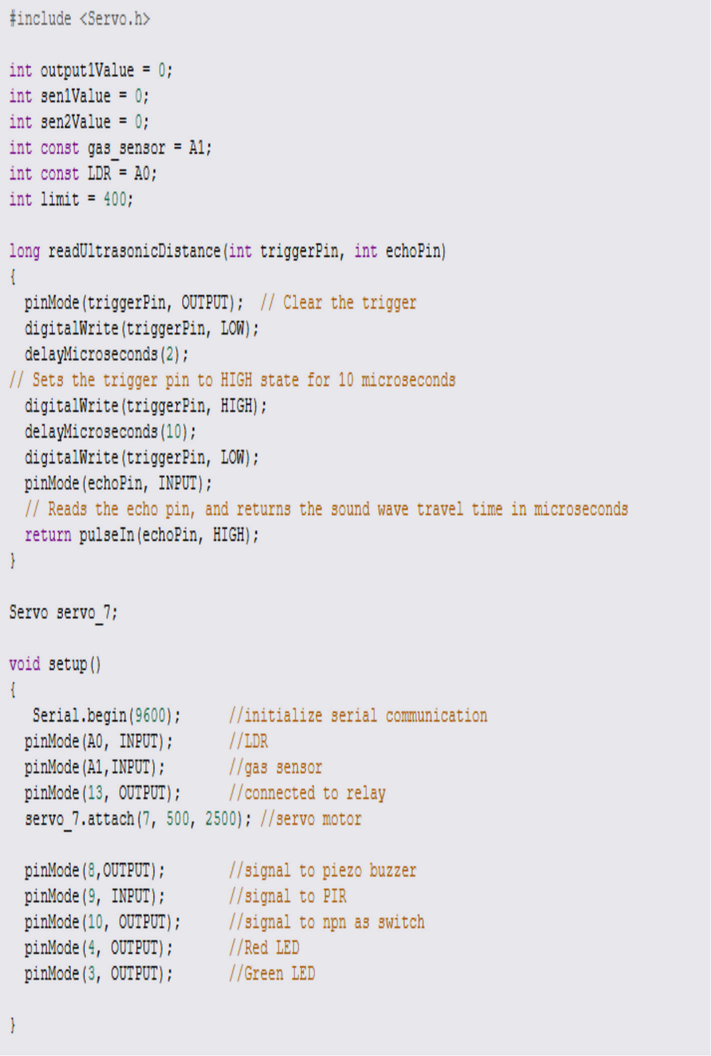
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN,DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("REAL TIME WEATHER STATISTICS: ");
  lcd.setCursor(0,1);
  lcd.print("WEATHER STATION");
  delay(2000);
  lcd.clear();

}
void loop() {
  delay(2000);
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  lcd.setCursor(0,0);
  lcd.print("Temperature: ");
  lcd.print(temperature);
  lcd.print("C");

  lcd.setCursor(0,1);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print("%");
  delay(5000);
  lcd.clear();

}
```



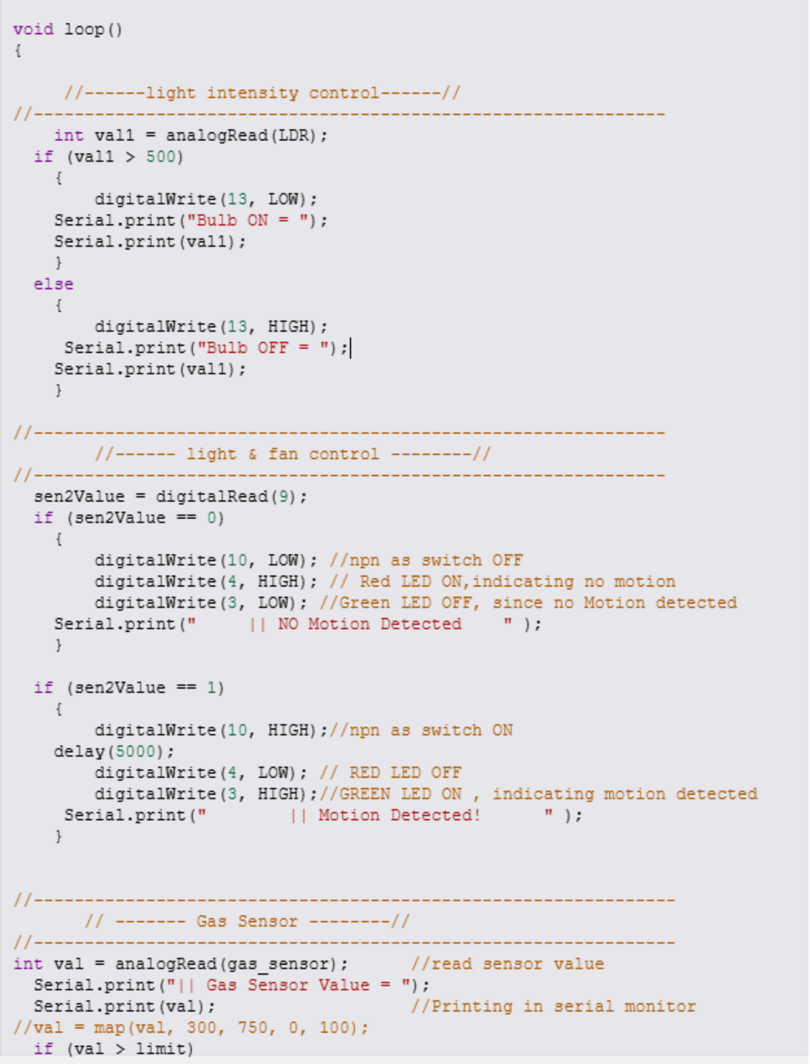


2 . Write an embedded C program to create a home automation system that simplifies daily routines by controlling devices remotely.

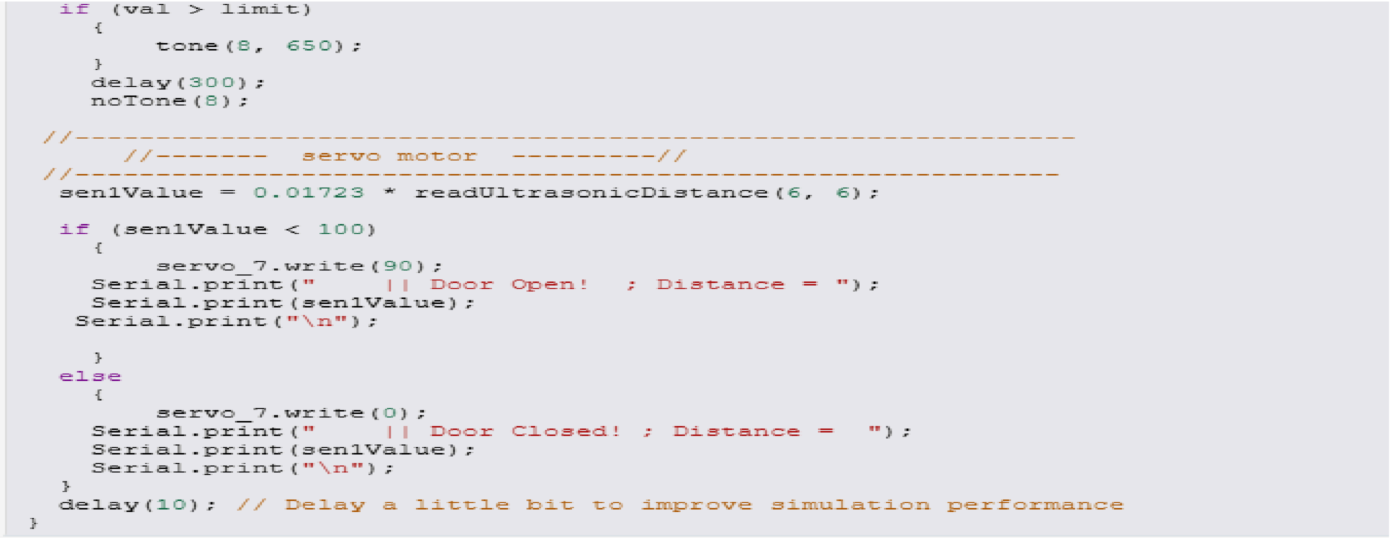




```
#include <Servo.h>

int output1Value = 0;
int sen1Value = 0;
int sen2Value = 0;
int const gas_sensor = A1;
int const LDR = A0;
int limit = 400;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
// Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

Servo servo_7;

void setup()
{
   Serial.begin(9600);      //initialize serial communication
  pinMode(A0, INPUT);       //LDR
  pinMode(A1,INPUT);        //gas sensor
  pinMode(13, OUTPUT);      //connected to relay
  servo_7.attach(7, 500, 2500); //servo motor

  pinMode(8,OUTPUT);        //signal to piezo buzzer
  pinMode(9, INPUT);        //signal to PIR
  pinMode(10, OUTPUT);      //signal to npn as switch
  pinMode(4, OUTPUT);       //Red LED
  pinMode(3, OUTPUT);       //Green LED


}
```

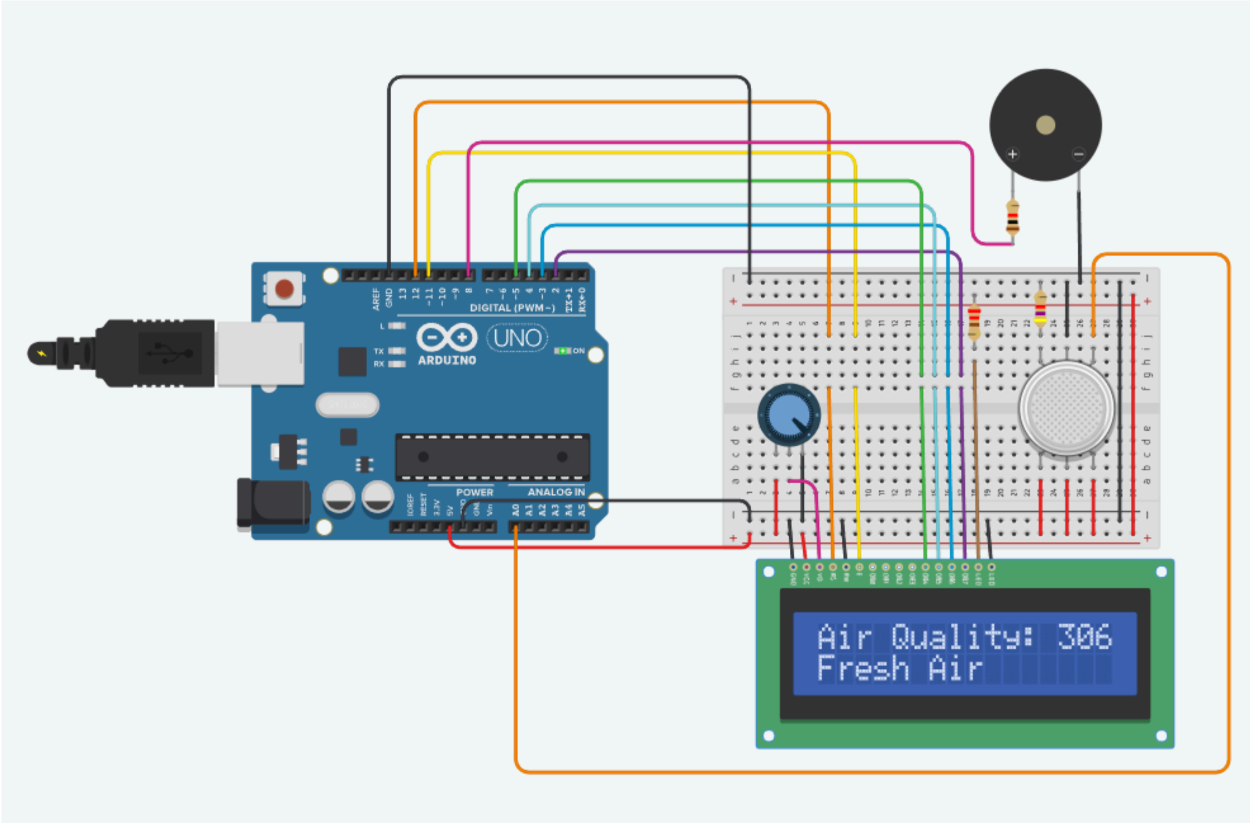


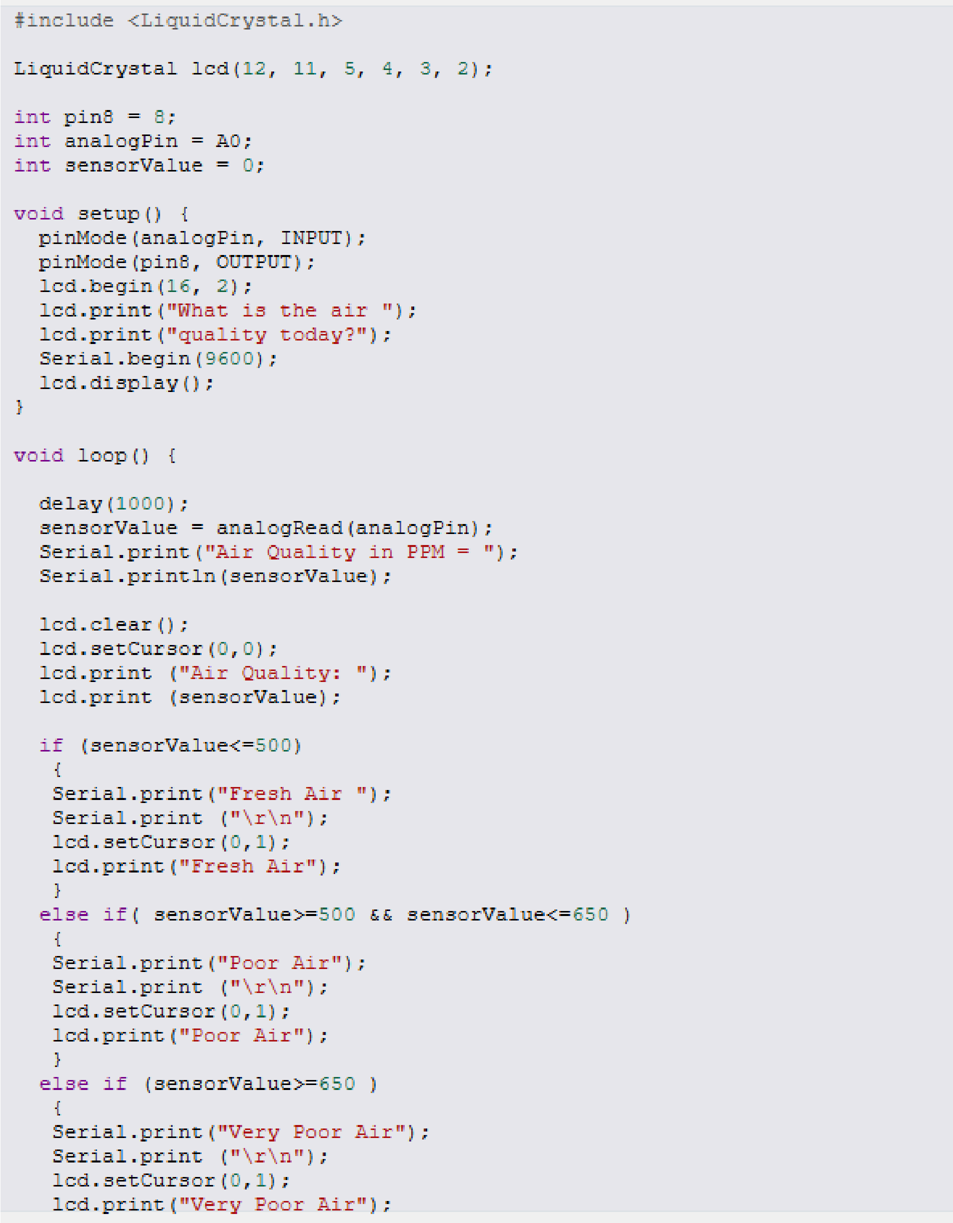
```
void loop()
{

     //------light intensity control------//
//--------------------------------------------------------------
    int val1 = analogRead(LDR);
  if (val1 > 500)
    {
        digitalWrite(13, LOW);
    Serial.print("Bulb ON = ");
    Serial.print(val1);
    }
  else
    {
        digitalWrite(13, HIGH);
     Serial.print("Bulb OFF = ");
    Serial.print(val1);
    }

//--------------------------------------------------------------
        //------ light & fan control --------//
//--------------------------------------------------------------
  sen2Value = digitalRead(9);
  if (sen2Value == 0)
    {
        digitalWrite(10, LOW); //npn as switch OFF
        digitalWrite(4, HIGH); // Red LED ON,indicating no motion
        digitalWrite(3, LOW); //Green LED OFF, since no Motion detected
    Serial.print("     || NO Motion Detected     " );
    }

  if (sen2Value == 1)
    {
        digitalWrite(10, HIGH);//npn as switch ON
    delay(5000);
        digitalWrite(4, LOW); // RED LED OFF
        digitalWrite(3, HIGH);//GREEN LED ON , indicating motion detected
     Serial.print("        || Motion Detected!      " );
    }


//---------------------------------------------------------------
       // ------- Gas Sensor --------//
//---------------------------------------------------------------
int val = analogRead(gas_sensor);      //read sensor value
  Serial.print("|| Gas Sensor Value = ");
  Serial.print(val);                   //Printing in serial monitor
//val = map(val, 300, 750, 0, 100);
  if (val > limit)
```



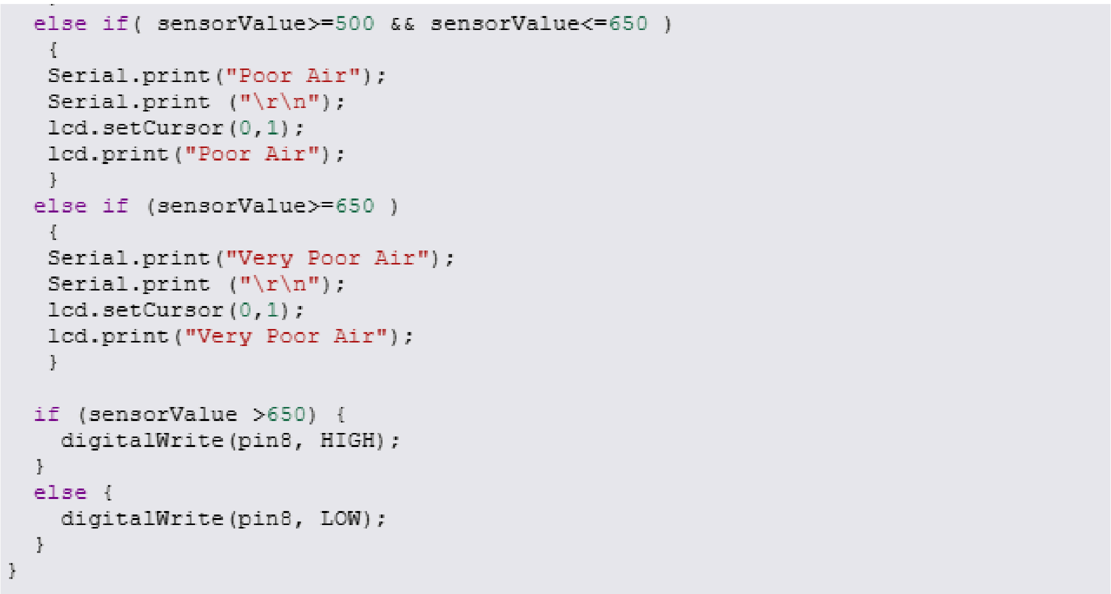
```
  if  (val  >  limit)
     {
         tone(8,  650);
     }
     delay(300);
     noTone(8);

 //---------------------------------------------------------------
      //-------   servo motor   ---------//
 //--------------------------------------------------------------
  sen1Value  =  0.01723  *  readUltrasonicDistance(6,  6);

  if  (sen1Value  <  100)
     {
         servo_7.write(90);
     Serial.print("     || Door Open!   ; Distance  =  ");
     Serial.print(sen1Value);
    Serial.print("\n");

     }
  else
     {
         servo_7.write(0);
     Serial.print("     || Door Closed! ; Distance  =   ");
     Serial.print(sen1Value);
     Serial.print("\n");
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```

3. Write a Embedded C Program to Create an Air Pollution Monitoring System that tracks air quality levels in real-time to ensure a healthier environment.



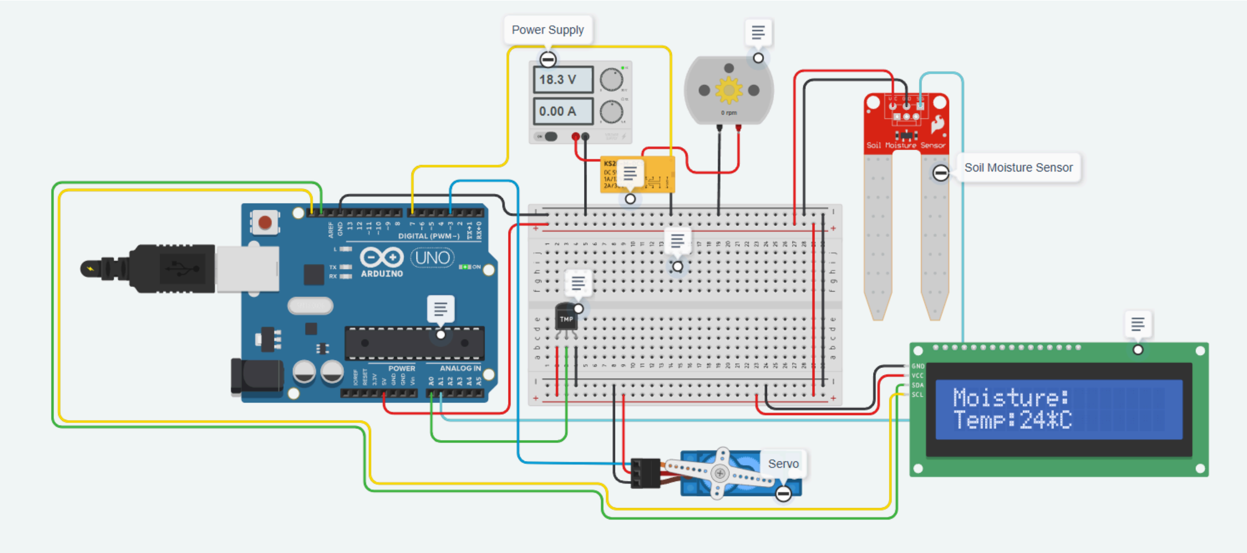


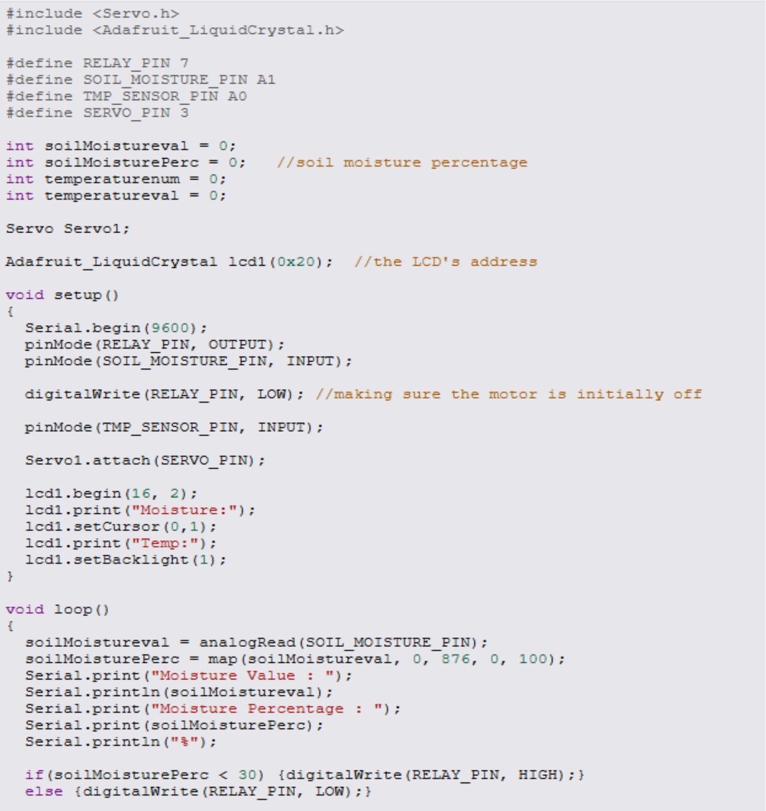
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int pin8 = 8;
int analogPin = A0;
int sensorValue = 0;

void setup() {
  pinMode(analogPin, INPUT);
  pinMode(pin8, OUTPUT);
  lcd.begin(16, 2);
  lcd.print("What is the air ");
  lcd.print("quality today?");
  Serial.begin(9600);
  lcd.display();
}

void loop() {

  delay(1000);
  sensorValue = analogRead(analogPin);
  Serial.print("Air Quality in PPM = ");
  Serial.println(sensorValue);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print ("Air Quality: ");
  lcd.print (sensorValue);

  if (sensorValue<=500)
   {
   Serial.print("Fresh Air ");
   Serial.print ("\r\n");
   lcd.setCursor(0,1);
   lcd.print("Fresh Air");
   }
  else if( sensorValue>=500 && sensorValue<=650 )
   {
   Serial.print("Poor Air");
   Serial.print ("\r\n");
   lcd.setCursor(0,1);
   lcd.print("Poor Air");
   }
  else if (sensorValue>=650 )
   {
   Serial.print("Very Poor Air");
   Serial.print ("\r\n");
   lcd.setCursor(0,1);
   lcd.print("Very Poor Air");
```



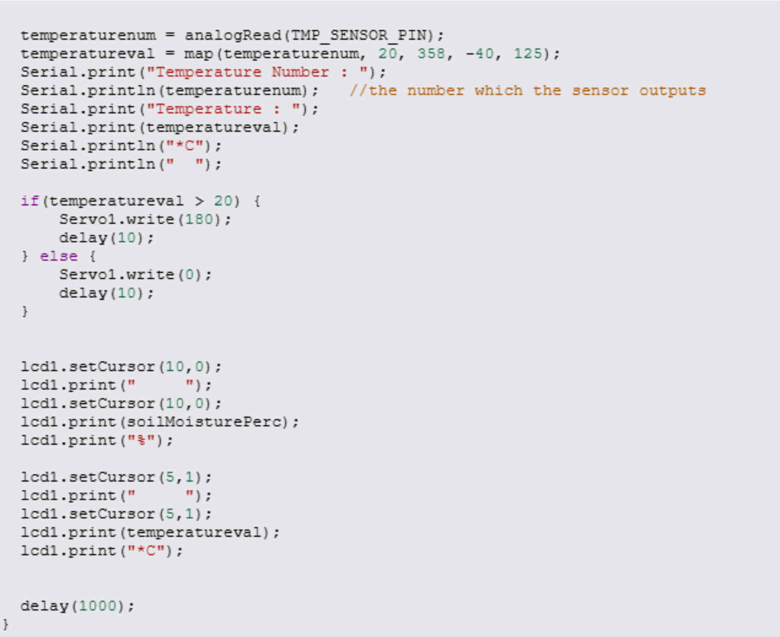
```
  else if( sensorValue>=500 && sensorValue<=650 )
   {
   Serial.print("Poor Air");
   Serial.print ("\r\n");
   lcd.setCursor(0,1);
   lcd.print("Poor Air");
   }
  else if (sensorValue>=650 )
   {
   Serial.print("Very Poor Air");
   Serial.print ("\r\n");
   lcd.setCursor(0,1);
   lcd.print("Very Poor Air");
   }

  if (sensorValue >650) {
    digitalWrite(pin8, HIGH);
  }
  else {
    digitalWrite(pin8, LOW);
  }
}
```

4. Write a Embedded C Program to Create an IoT-based Smart Irrigation System for Agriculture that automates watering based on weather and soil conditions



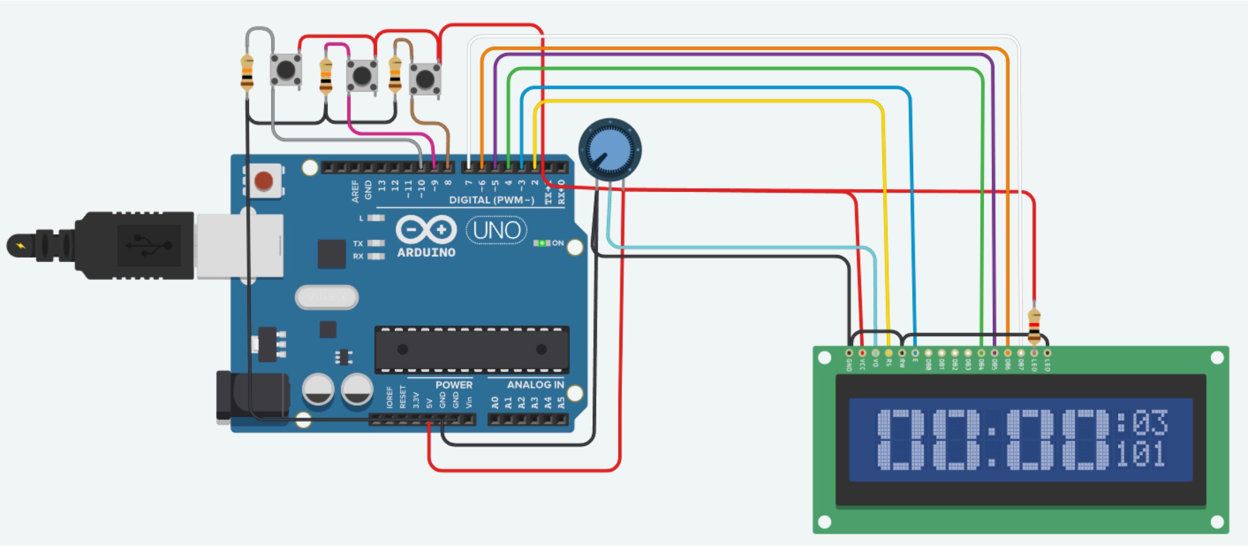


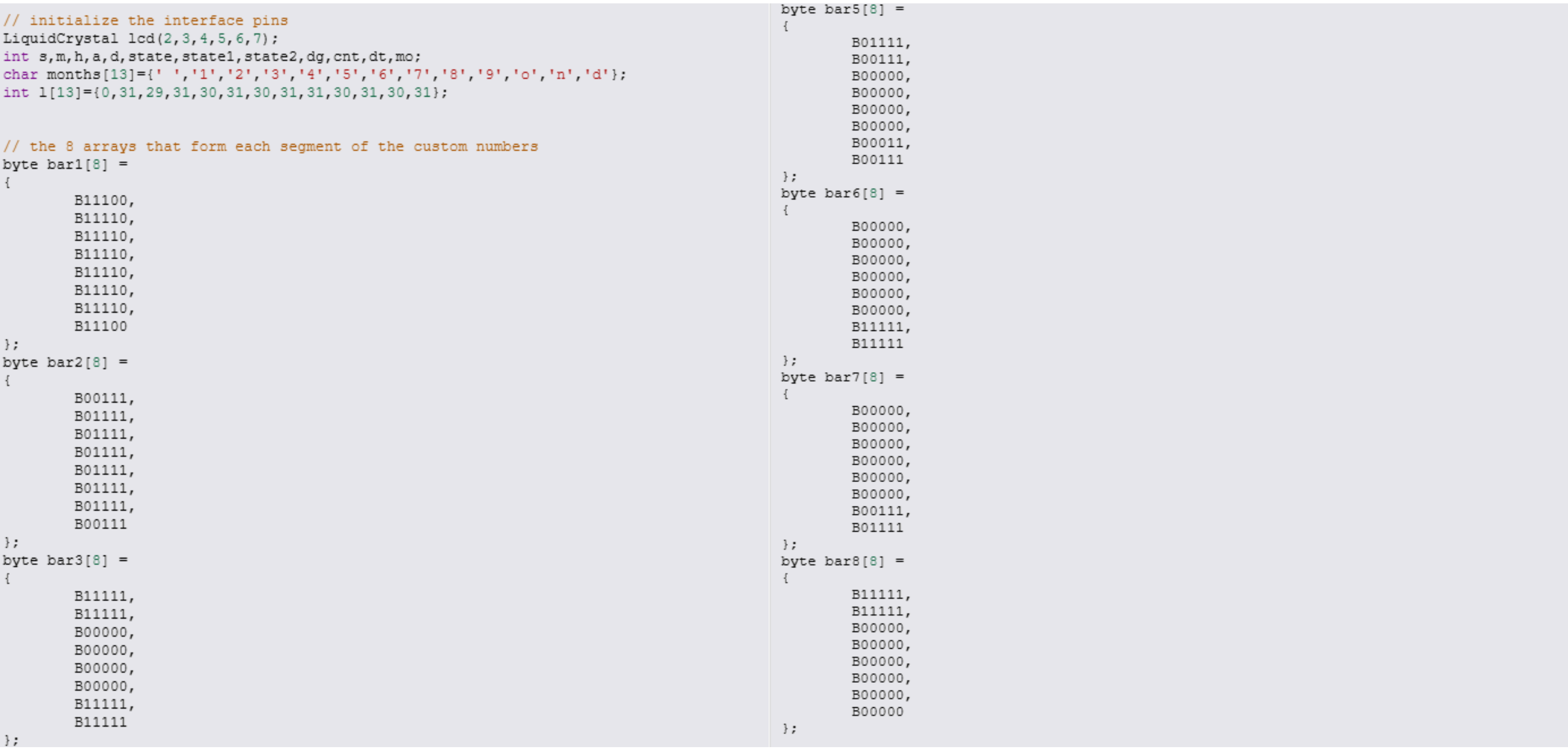
```
#include <Servo.h>
#include <Adafruit_LiquidCrystal.h>

#define RELAY_PIN 7
#define SOIL_MOISTURE_PIN A1
#define TMP_SENSOR_PIN A0
#define SERVO_PIN 3

int soilMoistureval = 0;
int soilMoisturePerc = 0;   //soil moisture percentage
int temperaturenum = 0;
int temperatureval = 0;

Servo Servo1;

Adafruit_LiquidCrystal lcd1(0x20);  //the LCD's address

void setup()
{
  Serial.begin(9600);
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(SOIL_MOISTURE_PIN, INPUT);

  digitalWrite(RELAY_PIN, LOW); //making sure the motor is initially off

  pinMode(TMP_SENSOR_PIN, INPUT);

  Servo1.attach(SERVO_PIN);

  lcd1.begin(16, 2);
  lcd1.print("Moisture:");
  lcd1.setCursor(0,1);
  lcd1.print("Temp:");
  lcd1.setBacklight(1);
}

void loop()
{
  soilMoistureval = analogRead(SOIL_MOISTURE_PIN);
  soilMoisturePerc = map(soilMoistureval, 0, 876, 0, 100);
  Serial.print("Moisture Value : ");
  Serial.println(soilMoistureval);
  Serial.print("Moisture Percentage : ");
  Serial.print(soilMoisturePerc);
  Serial.println("%");

  if(soilMoisturePerc < 30) {digitalWrite(RELAY_PIN, HIGH);}
  else {digitalWrite(RELAY_PIN, LOW);}
```



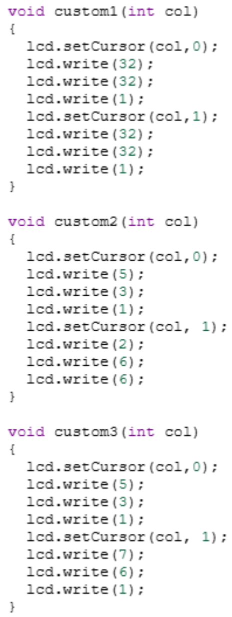
```
  temperaturenum = analogRead(TMP_SENSOR_PIN);
  temperatureval = map(temperaturenum, 20, 358, -40, 125);
  Serial.print("Temperature Number : ");
  Serial.println(temperaturenum);   //the number which the sensor outputs
  Serial.print("Temperature : ");
  Serial.print(temperatureval);
  Serial.println("*C");
  Serial.println("  ");

  if(temperatureval > 20) {
      Servo1.write(180);
      delay(10);
  } else {
      Servo1.write(0);
      delay(10);
  }


  lcd1.setCursor(10,0);
  lcd1.print("     ");
  lcd1.setCursor(10,0);
  lcd1.print(soilMoisturePerc);
  lcd1.print("%");

  lcd1.setCursor(5,1);
  lcd1.print("     ");
  lcd1.setCursor(5,1);
  lcd1.print(temperatureval);
  lcd1.print("*C");


  delay(1000);
}
```

5. Write a Embedded C Program to Create a Smart Alarm Clock that adjusts to your schedule and environment, waking you up intelligently.

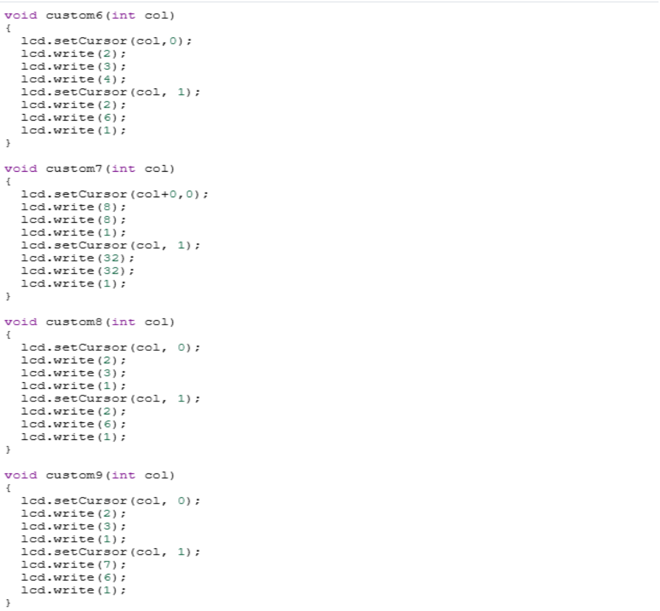




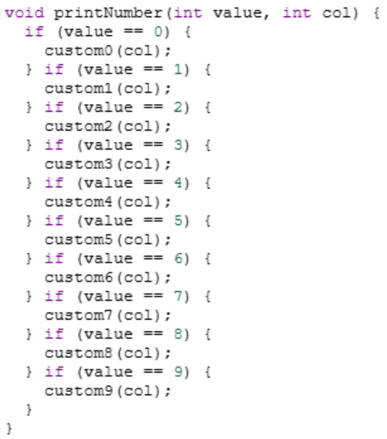
```
// initialize the interface pins
LiquidCrystal lcd(2,3,4,5,6,7);
int s,m,h,a,d,state,state1,state2,dg,cnt,dt,mo;
char months[13]={' ','1','2','3','4','5','6','7','8','9','o','n','d'};
int l[13]={0,31,29,31,30,31,30,31,31,30,31,30,31};


// the 8 arrays that form each segment of the custom numbers
byte bar1[8] =
{
        B11100,
        B11110,
        B11110,
        B11110,
        B11110,
        B11110,
        B11110,
        B11100
};
byte bar2[8] =
{
        B00111,
        B01111,
        B01111,
        B01111,
        B01111,
        B01111,
        B01111,
        B00111
};
byte bar3[8] =
{
        B11111,
        B11111,
        B00000,
        B00000,
        B00000,
        B00000,
        B11111,
        B11111
};
byte bar5[8] =
{
        B01111,
        B00111,
        B00000,
        B00000,
        B00000,
        B00000,
        B00011,
        B00111
};
byte bar6[8] =
{
        B00000,
        B00000,
        B00000,
        B00000,
        B00000,
        B00000,
        B11111,
        B11111
};
byte bar7[8] =
{
        B00000,
        B00000,
        B00000,
        B00000,
        B00000,
        B00000,
        B00111,
        B01111
};
byte bar8[8] =
{
        B11111,
        B11111,
        B00000,
        B00000,
        B00000,
        B00000,
        B00000,
        B00000
};
```



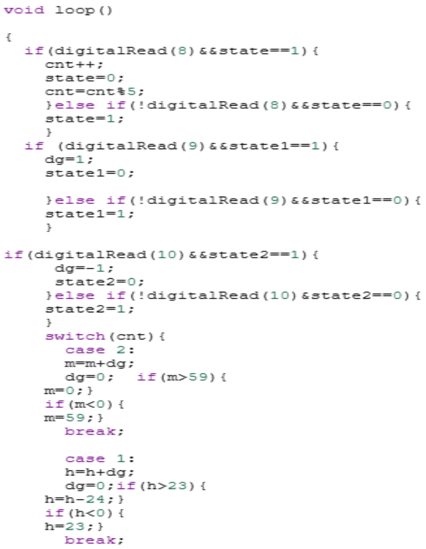
```
void custom1(int col)
{
  lcd.setCursor(col,0);
  lcd.write(32);
  lcd.write(32);
  lcd.write(1);
  lcd.setCursor(col,1);
  lcd.write(32);
  lcd.write(32);
  lcd.write(1);
}

void custom2(int col)
{
  lcd.setCursor(col,0);
  lcd.write(5);
  lcd.write(3);
  lcd.write(1);
  lcd.setCursor(col, 1);
  lcd.write(2);
  lcd.write(6);
  lcd.write(6);
}

void custom3(int col)
{
  lcd.setCursor(col,0);
  lcd.write(5);
  lcd.write(3);
  lcd.write(1);
  lcd.setCursor(col, 1);
  lcd.write(7);
  lcd.write(6);
  lcd.write(1);
}
```



```
void custom6(int col)
{
  lcd.setCursor(col,0);
  lcd.write(2);
  lcd.write(3);
  lcd.write(4);
  lcd.setCursor(col, 1);
  lcd.write(2);
  lcd.write(6);
  lcd.write(1);
}

void custom7(int col)
{
  lcd.setCursor(col+0,0);
  lcd.write(8);
  lcd.write(8);
  lcd.write(1);
  lcd.setCursor(col, 1);
  lcd.write(32);
  lcd.write(32);
  lcd.write(1);
}

void custom8(int col)
{
  lcd.setCursor(col, 0);
  lcd.write(2);
  lcd.write(3);
  lcd.write(1);
  lcd.setCursor(col, 1);
  lcd.write(2);
  lcd.write(6);
  lcd.write(1);
}

void custom9(int col)
{
  lcd.setCursor(col, 0);
  lcd.write(2);
  lcd.write(3);
  lcd.write(1);
  lcd.setCursor(col, 1);
  lcd.write(7);
  lcd.write(6);
  lcd.write(1);
}
```



```
void printNumber(int value, int col) {
  if (value == 0) {
    custom0(col);
  } if (value == 1) {
    custom1(col);
  } if (value == 2) {
    custom2(col);
  } if (value == 3) {
    custom3(col);
  } if (value == 4) {
    custom4(col);
  } if (value == 5) {
    custom5(col);
  } if (value == 6) {
    custom6(col);
  } if (value == 7) {
    custom7(col);
  } if (value == 8) {
    custom8(col);
  } if (value == 9) {
    custom9(col);
  }
}
```



```
void loop()

{
  if(digitalRead(8)&&state==1){
    cnt++;
    state=0;
    cnt=cnt%5;
    }else if(!digitalRead(8)&&state==0){
    state=1;
    }
  if (digitalRead(9)&&state1==1){
    dg=1;
    state1=0;

    }else if(!digitalRead(9)&&state1==0){
    state1=1;
    }

if(digitalRead(10)&&state2==1){
     dg=-1;
     state2=0;
    }else if(!digitalRead(10)&state2==0){
    state2=1;
    }
    switch(cnt){
      case 2:
      m=m+dg;
      dg=0;   if(m>59){
    m=0;}
    if(m<0){
    m=59;}
      break;

      case 1:
      h=h+dg;
      dg=0;if(h>23){
    h=h-24;}
    if(h<0){
    h=23;}
      break;
h=h;
d=(h)%10;
printNumber(d, 3);
d=(h)/10;
 printNumber(d, 0);

     d=m%10;
   printNumber(d, 10);
   d=m/10;
   printNumber(d, 7);
   lcd.setCursor(14, 0);


   lcd.print(s/10);

   lcd.print(s%10);

   lcd.setCursor(13, 1);
    lcd.print(months[mo]);
   lcd.print(dt/10);

   lcd.print(dt%10);
      if(cnt==0){
         s++;
        lcd.setCursor(6, 0);
    lcd.print(" ");
     lcd.setCursor(6, 1);
    lcd.print(" ");
     lcd.setCursor(13,0);
             lcd.print(" ");
    delay(500);
  lcd.setCursor(6, 0);
    lcd.print(".");
     lcd.setCursor(6, 1);
    lcd.print(".");
      lcd.setCursor(13,0);
    lcd.print(":");
    delay(500);
     }}
```