

3a)

```
int LEDpin = 11;

void setup()
{ pinMode(LEDpin, OUTPUT);
}

void loop()
{
digitalWrite(LEDpin, HIGH);

delay(500);

digitalWrite(LEDpin, LOW);

delay(500);

}
```

---

3b)

```
int red = 7; int yellow = 5; int green = 3; void setup() { pinMode(red, OUTPUT); pinMode(yellow,
OUTPUT); pinMode(green, OUTPUT); } void loop() { digitalWrite(red, HIGH); delay(15000);
digitalWrite(red, LOW); for (int i = 0; i < 6; i++) { digitalWrite(yellow, HIGH); delay(1000);
digitalWrite(yellow, LOW); delay(500); } digitalWrite(green, HIGH); delay(20000); digitalWrite(green,
LOW); for (int i = 0; i < 6; i++) { digitalWrite(yellow, HIGH); delay(1000); digitalWrite(yellow, LOW);
delay(500); } }
```

---

3c)

```
const int buzzer = 8; const int trig_pin = 9; const int echo_pin = 10; float timing = 0.0; float distance =
0.0; void setup(){ pinMode(echo_pin, INPUT); pinMode(trig_pin, OUTPUT); pinMode(buzzer,
OUTPUT); digitalWrite(trig_pin, LOW); digitalWrite(buzzer, LOW); Serial.begin(9600); } void loop(){
digitalWrite(trig_pin, LOW); delay(2); digitalWrite(trig_pin, HIGH); delay(10); digitalWrite(trig_pin,
LOW); timing = PulseIn_echopin(HIGH); distance = (timing*0.034) / 2; Serial.print("Distance: ");
Serial.print(distance); Serial.print("cm | "); Serial.print(distance /2.54); Serial.println("in"); if
(distance <= 50) { tone(buzzer, 500); } else { noTone(buzzer); } delay(100); }
```

---

3d)

```
int IRSensor = 9; int LED = 13; void setup() { Serial.begin(115200); Serial.println("Serial Working");
pinMode(IRSensor, INPUT); pinMode(LED, OUTPUT); } void loop() { int sensorStatus =
digitalRead(IRSensor); if (sensorStatus == 1) { digitalWrite(LED, LOW); Serial.println("Motion
Ended!"); } else { digitalWrite(LED, HIGH); Serial.println("Motion Detected!"); } }
```

---

3e)

```
int sensor_pin = A0; int output_value ; void setup() { Serial.begin(9600); Serial.println("Reading From
the Sensor ..."); delay(2000); } void loop() { output_value= analogRead(sensor_pin); output_value =
map(output_value,550,0,0,100); Serial.print("Mositure : "); Serial.print(output_value);
Serial.println("%"); delay(1000); }
```

---

3f)

```
#include "DHT.h" #define DHT11_PIN 2 DHT dht11(DHT11_PIN, DHT11); void setup() dht11.begin(); {
Serial.begin(9600); } void loop() { delay(2000); float humi = dht11.readHumidity(); float tempC =
dht11.readTemperature(); float tempF = dht11.readTemperature(true); if (isnan(humi) ||
isnan(tempC) || isnan(tempF)) { Serial.println("Failed to read from DHT11 sensor!"); } else {
Serial.print("DHT11# Humidity: "); Serial.print(humi); Serial.print("%"); Serial.print(" | ");
Serial.print("Temperature: "); Serial.print(tempC); Serial.print("°C ~ "); Serial.print(tempF);
Serial.println("°F"); } }
```

---

3g)

```
#include sbit red = P1^0; sbit yellow = P1^1; sbit green = P1^2; void main() { unsigned int i; P1 =
0x00; P2 = 0x00; while (1) { red = 1; yellow = 0; green = 0; for (i = 0; i < 60000; i++); // Delay red = 0;
yellow = 1; green = 0; for (i = 0; i < 60000; i++); // Delay red = 0; yellow = 0; green = 1; for (i = 0; i <
60000; i++); // Delay } }
```

---

3h)

```
#include sbit LED=P2^3; void delay_ms(unsigned int ms_count) {
unsigned int i,j; for(i=0;i<ms_count,i++)
}
}

void main(void)
{
while(1)
{
LED=~LED;
delay_ms(500);
}
}
```

---

5)

```
#include<dht.h>

#define dht_dpina A1 //no ; here. Set equal to channel sensor is on

dht DHT;

void setup()
{
Serial.begin(9600);
delay(300);
```

```

Serial.println("Humidity and temperature\n\n");
delay(700);
}
void loop()
{
DHT.read11(dht_dpin);
Serial.print("Current humidity = ");
Serial.print(DHT.humidity);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(DHT.temperature);
Serial.println("C ");
delay(800); seconds, fastest,}

```

#### Python Code

```

import time import serial import urllib2 ser = serial.Serial('COM7', 9600)
while True: message = ser.readline() print(message)

#print(message[19:21])#print(message[41:43]) response =
urllib2.urlopen('https://api.thingspeak.com/update?api_key=apikey&fie
ld1='+message[19:21]+'&field2='+message[41:43]) html = response.read()

time.sleep(0.5)

```