ID: 11461

NAME : ASFANDYAR AWAIS

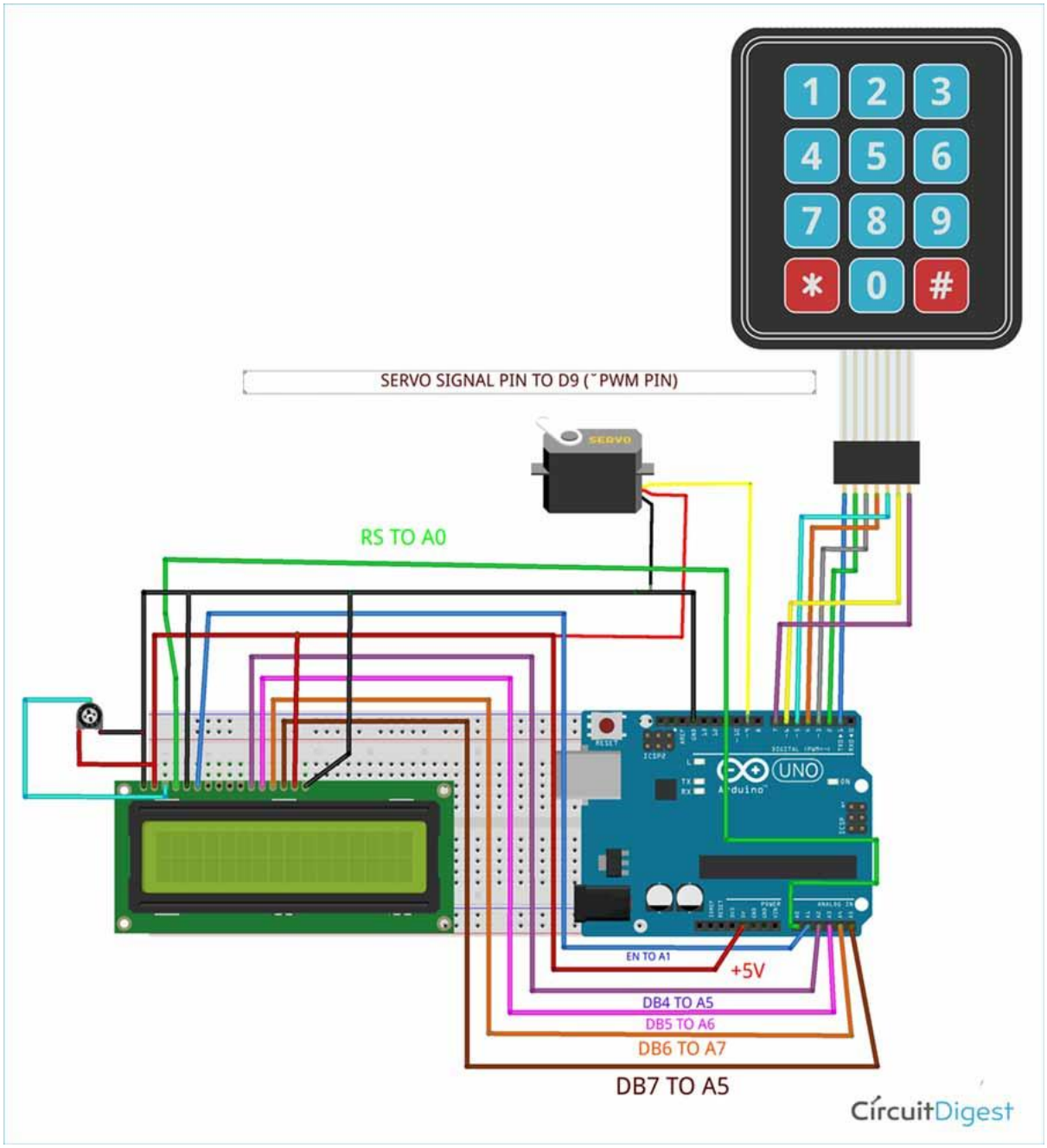
INSTRUCTOR : ENGINEER WAQAS

ANSWER Q1:

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>
Servo myservo;
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
#define Password_Lenght 7 // Give enough room for six chars + NULL char
int pos = 0;    // variable to store the servo position
char Data[Password_Lenght]; // 6 is the number of chars it can hold + the null char = 7
char Master[Password_Lenght] = "55445";
byte data_count = 0, master_count = 0;
bool Pass_is_good;
char customKey;
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
bool door = true;
byte rowPins[ROWS] = {1, 2, 3, 4}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 6, 7}; //connect to the column pinouts of the keypad
Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS); //initialize
an instance of class NewKeypad
void setup()
{
  myservo.attach(9);
```

```
 ServoClose();
 lcd.begin(16, 2);
 lcd.print("  Arduino  Door");
 lcd.setCursor(0, 1);
 lcd.print("--Look  project--");
 delay(3000);
 lcd.clear();
}
void loop()
{
 if (door == 0)
 {
  customKey  = customKeypad.getKey();
  if (customKey  == '#')
  {
   lcd.clear();
   ServoClose();
   lcd.print("   Door is close");
   delay(3000);
   door = 1;
  }
 }
 else Open();
}
void clearData()
{
 while  (data_count  != 0)
 { // This  can be used  for any  array size,
  Data[data_count--]  = 0; //clear  array  for new data
 }
 return;
}
void ServoOpen()
{
 for (pos = 180; pos >= 0; pos -= 5) { // goes from 0 degrees to 180 degrees
  // in steps of 1 degree
  myservo.write(pos);              // tell  servo to go to position  in variable  'pos'
  delay(15);                       // waits  15ms for the servo to reach the position
 }
}
void ServoClose()
{
 for (pos = 0; pos <= 180; pos += 5) { // goes from 180 degrees to 0 degrees
  myservo.write(pos);              // tell  servo to go to position  in variable  'pos'
  delay(15);                       // waits  15ms for the servo to reach the position
 }
```

```
}
void Open()
{
 lcd.setCursor(0, 0);
 lcd.print("  Enter Password");
 customKey = customKeypad.getKey();
 if (customKey) // makes sure a key is actually pressed, equal to (customKey != NO_KEY)
 {
  Data[data_count] = customKey; // store char into data array
  lcd.setCursor(data_count, 1); // move cursor to show each new char
  lcd.print(Data[data_count]); // print char at said cursor
  data_count++; // increment data array by 1 to store new char, also keep track of the number
of chars entered
 }
 if (data_count == Password_Lenght - 1) // if the array index is equal to the number of
expected chars, compare data to master
 {
  if (!strcmp(Data, Master)) // equal to (strcmp(Data, Master) == 0)
  {
    lcd.clear();
    ServoOpen();
    lcd.print("   Door is Open");
    door = 0;
  }
    lcd.clear();
    lcd.print("   Wrong Password");
    delay(1000);
    door = 1;
  }
  clearData();
 }
  else
  {}
```



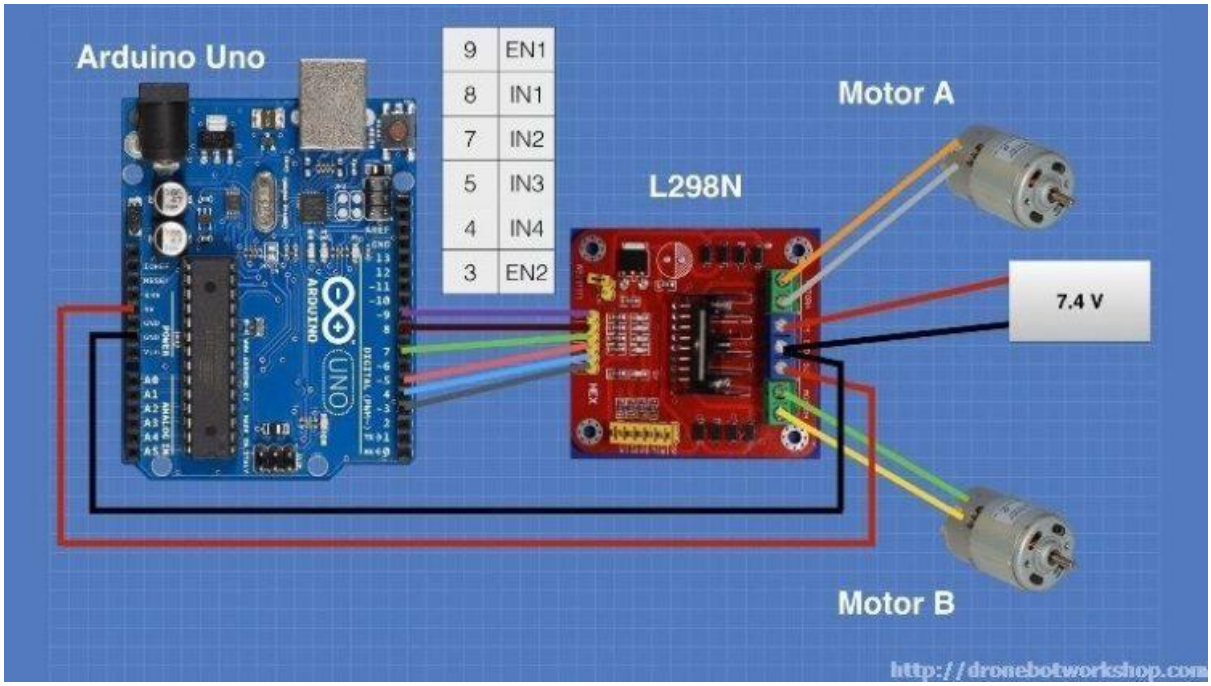
ANSWER Q2:

```
#include

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer

int val;    // variable to read the value from the analog pin
```

```
void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
void loop()
{
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 an
d 1023)
  val = map(val, 0, 1023,0, 45);     // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
}
#include "MotorDriver.h"
const int POT_PIN = A0;
const int POT_PIN2 = A1;
int motorSpeed = 0;
int potVal = 0;
int pinI1=8;//define I1 interface
int pinI2=11;//define I2 interface
int pinI3=12;
int pinI4=13;
int speedpinA=9;//enable motor A
int speedpinB=10;
void setup()
```

```
{
Serial.begin(9600);
TCCR1B = TCCR1B & 0b11111000 | 0x01;
pinMode(pinI1,OUTPUT);
pinMode(pinI2,OUTPUT);
pinMode(pinI3,OUTPUT);
pinMode(pinI4,OUTPUT);
pinMode(speedpinA,OUTPUT);
pinMode(speedpinB,OUTPUT);
}
void loop()
{
potVal = analogRead(POT_PIN);
potVal = analogRead(POT_PIN2);
motorSpeed = map(potVal, 0, 1023, 0, 255);
Serial.print(potVal);
Serial.print(motorSpeed);
Serial.println();
analogWrite(speedpinA, motorSpeed);
analogWrite(speedpinB, motorSpeed);
digitalWrite(pinI2,LOW);//turn DC Motor A move anticlockwise
digitalWrite(pinI1,HIGH);
digitalWrite(pinI4,LOW);
```

```
digitalWrite(pinI3,HIGH);

}
```



ANSWER Q3:

```
int L1 = 13;
int L2 = 12;
int L3 = 11;
int L4 = 10;
int L5 = 9;
int L6 = 8;
int L7 = 7; //7 LED pin

int buttonPin = 6; //the number of the pushbutton pin

int de=50; // delay time
```

```
int p=0;    // variable  for pattem

int buttonState  = 0;         // variable  for reading  the pushbutton  status

void setup()  {


pinMode(L1,  OUTPUT);
pinMode(L2,  OUTPUT);
pinMode(L3,  OUTPUT);
pinMode(L4,  OUTPUT);
pinMode(L5,  OUTPUT);
pinMode(L6,  OUTPUT);
pinMode(L7,  OUTPUT);

pinMode(buttonPin,  INPUT);


}

void loop()
{
buttonState  = digitalRead(buttonPin);

if (buttonState  == HIGH)

{
p++;
delay(2000);
}
```

```
if(p==1)
{
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);   //1
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);   //2
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);   //3
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
```

```
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);   //4
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);   //5
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
    digitalWrite(L7,0);   //6
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,1);   //7
```

```
delay(de);
}




if(p==2)
{
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);   //7
delay(de);

digitalWrite(L1,0)
    digitalWrite(L2,0);
digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,1);
    digitalWrite(L7,0);   //6
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,1);
```

```
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //5
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0;
digitalWrite(L6,0);
    digitalWrite(L7,0);   //4
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,1);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //3
    delay(de);


    digitalWrite(L1,0);
    digitalWrite(L2,1);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
digitalWrite(L6,0);
    digitalWrite(L7,0);   //2
    delay(de);
```

```
digitalWrite(L1,1);
digitalWrite(L2,0;
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);    //1
delay(de);

}

if(p==3)
{
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);    //1
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);    //2
delay(de);
```

```
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,1);
digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //3
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //4
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,1);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //5
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
```

```
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);   //6
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);   //7
delay(de);



digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);   //6
delay(de);

digitalWrite(L1,);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
```

```
    digitalWrite(L7,0);   //5
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
digitalWrite(L7,0);   //4
    delay(de);

    digitalWrite(L1,0);
digitalWrite(L2,0);
    digitalWrite(L3,1);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //3
    delay(de);


    digitalWrite(L1,0);
    digitalWrite(L2,1);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //2
    delay(de);
    }
```

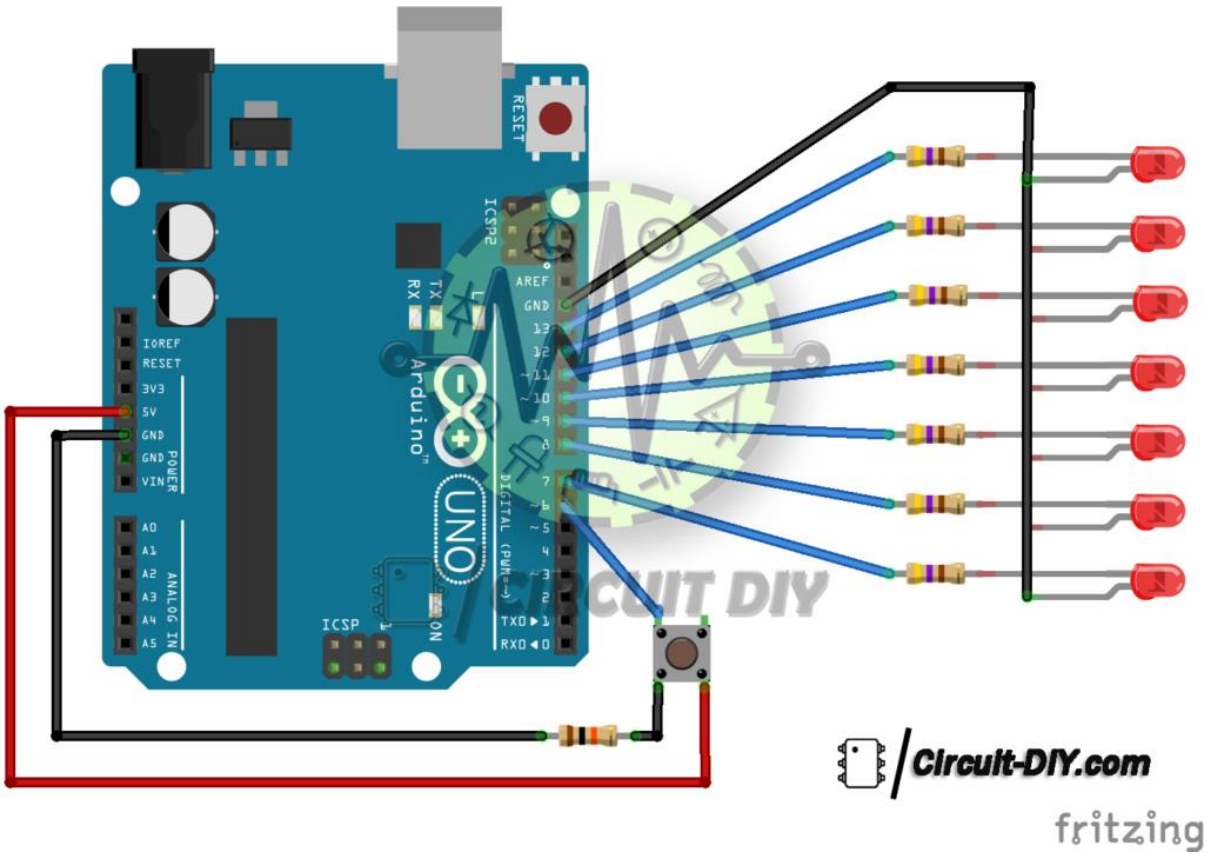
```
if(p==4)
{
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);   //1,7
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);   //2,6
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);   //3,5
```

```
    delay(de);


    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0);
digitalWrite(L6,0);
    digitalWrite(L7,0);   //4
    delay(de);


    }



    if(p==5)
    {


    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0);   //4
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,1);
    digitalWrite(L4,0);
```

```
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);   //3,5
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);   //2,6
delay(de);

digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);   //1,7
delay(de);
}

if(p==6)
{
digitalWrite(L1,1);
delay(de);
digitalWrite(L2,1);
delay(de);
digitalWrite(L3,1);
```

```
delay(de);
digitalWrite(L4,1);
delay(de);
digitalWrite(L5,1);
delay(de);
digitalWrite(L6,1);
delay(de);
digitalWrite(L7,1);   //1,7
delay(de);
digitalWrite(L7,0);   //1,7
delay(de);
digitalWrite(L6,0);
delay(de);
digitalWrite(L5,0);
delay(de);
digitalWrite(L4,0);
delay(de);
digitalWrite(L3,0);
delay(de);
digitalWrite(L2,0);
delay(de);
digitalWrite(L1,0);
delay(de);



}


if(p==7)
{
digitalWrite(L1,0);
digitalWrite(L2,0);
```

```
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);   //1,7
p=0;
}

}
```



ANSWER Q4:

```
byte img1[ ] = {
   B01100110,
   B11111111,
   B11111111,
   B11111111,
```

```
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte img2[] = {
  B00000000,
  B01100110,
  B01100110,
  B00000000,
  B00011000,
  B10000001,
  B01000010,
  B00111100
};

const int rowPins[] = {13,12,11,10,9,8,7,6};
const int columnPins[] = {5,4,3,2,14,15,16,17};

const int but1 = 18;
const int but2 = 19;

int but1State = 0;
int but2State = 0;
int but1Now;
int but2Now;

int value1;
int value2;

void setup() {

  for (int i = 0; i<8; i++){
    pinMode(rowPins[i], OUTPUT);
    pinMode(columnPins[i], OUTPUT);
    digitalWrite(columnPins[i], HIGH);
  }

  pinMode(but1, INPUT);
  pinMode(but2, INPUT);

  but1State = digitalRead(but1);
  but2State = digitalRead(but2);
}

void loop() {

  but1Now = digitalRead(but1);
  delay(5);
```

```
  int but1Now1 = digitalRead(but1);

  if(but1Now  == but1Now1)
  {
    if(but1Now  != but1State)
    {
      value1  = 1;
      value2  = 0;
      //show(img2);
    }

    but1State = but1Now;
  }

  but2Now = digitalRead(but2);
  delay(5);
  int but2Now1 = digitalRead(but2);

  if(but2Now  == but2Now1)
  {
    if(but2Now  != but2State)
    {
      value1  = 0;
      value2  = 1;
      //show(img2);
    }

    but2State = but2Now;
  }

  if(value1  == 1)
  {
    show(img1);
  }

  if(value2  == 1)
  {
    show(img2);
  }
}

void show(byte*image)  {

  for(int  row = 0; row<8; row++)
  {
    digitalWrite(rowPins[row],  HIGH);

    for(int  col=0; col<8; col++)
    {
      boolean pixel = bitRead(image[row],col);
```

```
            if(pixel  == 1)
            {
               digitalWrite(columnPins[col],  LOW);
            }

            delayMicroseconds(150);
            digitalWrite(columnPins[col],  HIGH);
         }

         digitalWrite(rowPins[row],  LOW);
      }
}
```