```
// Includes the Servo library
#include <Servo.h>.
// Defines Tirg and Echo pins of the Ultrasonic Sensor
const int trigPin = 10;
const int echoPin = 11;
// Variables for the duration and the distance
long duration;
int distance;
int degree;
Servo myServo; // Creates a servo object for controlling the servo motor

//Define pins for the LED lights
const int LED1 = 2;
const int LED2 = 3;
const int LED3 = 4;
const int LED4 = 5;
const int LED5 = 6;
const int LED6 = 7;
const int LED7 = 8;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

// Sets the LEDs as Outputs
  pinMode(LED1 , OUTPUT);
  pinMode(LED2 , OUTPUT);
  pinMode(LED3 , OUTPUT);
  pinMode(LED4 , OUTPUT);
  pinMode(LED5 , OUTPUT);
  pinMode(LED6 , OUTPUT);
  pinMode(LED7 , OUTPUT);


  Serial.begin(9600);
  myServo.attach(12); // Defines on which pin is the servo motor attached
}


void loop() {
  // rotates the servo motor from 15 to 165 degrees
  for(degree=15;degree<=165;degree++){
  myServo.write(degree);
  delay(30);
```

```
  distance = calculateDistance();// Calls a function for calculating the distance
measured by the Ultrasonic sensor for each degree


  Serial.print(degree); // Sends the current degree into the Serial Port
  Serial.print(","); // Sends addition character right next to the previous value
needed later in the Processing IDE for indexing
  Serial.print(distance); // Sends the distance value into the Serial Port
  Serial.print("."); // Sends addition character right next to the previous value
needed later in the Processing IDE for indexing
  }

  // Repeats the previous lines from 165 to 15 degrees
  for(int degree=165;degree>15;degree--){
  myServo.write(degree);
  delay(30);
  distance = calculateDistance();
  Serial.print(degree);
  Serial.print(",");
  Serial.print(distance);
  Serial.print(".");
  }


}
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave
travel time in microseconds
  distance= duration*0.034/2;
//Prints the real distance in Arduino to make sure the beforementioned code works.
  Serial.print("the distance is: ");
  Serial.println(distance);

//Defines the distance to the point of illuminating LED lights
if ( distance <= 5 )
  {
    digitalWrite(LED1, HIGH);
```

```
}
else
{
  digitalWrite(LED1, LOW);
}
if ( distance <= 10 )
{
  digitalWrite(LED2, HIGH);
}
else
{
  digitalWrite(LED2, LOW);
}
if ( distance <= 15 )
{
   digitalWrite (LED3, HIGH);
}
else
{
  digitalWrite(LED3, LOW);
}
if ( distance <= 20 )
{
  digitalWrite(LED4, HIGH);
}
else
{
  digitalWrite(LED4, LOW);
}
if ( distance <= 25 )
{
  digitalWrite(LED5, HIGH);
}
else
{
  digitalWrite(LED5, LOW);
}
if ( distance <= 30 )
{
  digitalWrite(LED6, HIGH);
}
else
{
  digitalWrite(LED6, LOW);
}
```

```
  if ( distance <= 35 )
  {
    digitalWrite(LED7, HIGH);
  }
  else
  {
    digitalWrite(LED7, LOW);
  }

  return distance;
}
```