```
int speakerPin = 9;
const int trigPin = 2;
const int echoPin = 4;
const int potent = A0;

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}

void playNote(char note, int duration) {
  long names[] = { 0, 1, 2, 3, 4, 6, 7 };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 }; //frequency for the notes

  // play the tone corresponding to the note name
  for (int i = 0; i < 8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration);
    }
  }
}



void setup() {
  // begin serial communication:
  Serial.begin(9600);
  pinMode (speakerPin, OUTPUT);


}

void loop() {
 // establish variables for duration of the ping and the distance result in inches
  long duration, inches, cm;
```

```
  // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);


  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);


  // convert the time into a distance
  inches = microsecondsToInches(duration/2); //converts the microseconds from the ping to
inches
  cm = microsecondsToCentimeters(duration/2); // converts the microseconds from the pig
to centimeters


      playNote(inches, analogRead(A0));

  }
long microsecondsToInches(long microseconds)
{

  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
```

```
}
```