

[illegible]

```

//Note LED Variables
int dataPinNotes = 4;
int latchPinNotes = 5;
int clockPinNotes = 6;
byte LEDNotes = 0;

//Octive LED Variables
int dataPinOctive = 8;
int latchPinOctive = 9;
int clockPinOctive = 10;
byte LEDOctive =0;

//Var used to set the bit in the Byte for the shift reg to light the correct LED
int q = 0; // A No Representing the note to be played
int r = 0; // A no representing the bit which must be set for q to play
float j = 0; // A number used to determine which LED must be lit for a particular Note = <noOfFreq/7.
00 as in 6th line of active code in loop

void setup() {
  pinMode(buzzerPin, OUTPUT); //Set Pins as OUTPUT
  pinMode(dataPinNotes, OUTPUT);
  pinMode(latchPinNotes, OUTPUT);
  pinMode(clockPinNotes, OUTPUT);
  pinMode(dataPinOctive, OUTPUT);
  pinMode(latchPinOctive, OUTPUT);
  pinMode(clockPinOctive, OUTPUT);

```

```

Serial.begin (9600);
}

void loop() {

    if (playLoop == false) {

        for (int i = 0; i < noOfFreq; i++) {                                //Using the Tone Function with pin No 3
(buzzerPin), freq[] and toneDuration
            tone(buzzerPin, freq[i], freqDuration[i]*tempo);
            Serial.print("  Play Freq with tone function using freq and duration = ");
            Serial.print(freq[i]);
            Serial.print(" and ");
            Serial.println(freqDuration[i]*tempo);
            delay(freqDuration[i]*200);
            float j = i/7.00;
            q = (int)((j - (int)j)*10);
            updateToneLEDs();
            /*Serial.print("  j = ");
            Serial.println(j);*/
            updateOactiveLEDs(j);
        }
        noTone(3);
        delay(tempo*2);
    }

    else {
        for (int i = 0; i < noOfTones; i++) {                                //Using a loop (playTone) with pin
High and LOW with a delay

```

```

        /*Serial.print ("In Loop's for next loop with i = ");
        Serial.println (i);*/
playTone(tones[i], toneDuration[i]*tempo);
//pause between notes
delay(tempo/2);
float j = i/7.00;
q = (int)((j - (int)j)*10);
updateToneLEDs();
updateOctiveLEDs(j);

    }
    delay(tempo*5);
}
}

```

```

void playTone(int toneToPlay, int duration){ //Routine to play a tone given the Time High (toneTo
Play) in ms and duration of note also ms
/*Serial.print(" #playTone with toneToPlay and duration = ");
Serial.print(toneToPlay);
Serial.print(" and ");
Serial.println(duration);*/
for (long i = 0; i < duration * 2000L; i += toneToPlay*2) {
    digitalWrite (buzzerPin, HIGH);
    delayMicroseconds(toneToPlay);
    digitalWrite (buzzerPin, LOW);
    delayMicroseconds(toneToPlay);
}
}
}

```

```
void updateToneLEDs() {
    LEDNotes = 0;
    updateShiftReg(latchPinNotes, clockPinNotes, dataPinNotes, LEDNotes);
    /*Serial.print ("In 'updateToneLEDs' with latchPin, clockPin, dataPin and LEDNotes being ");
    Serial.print (latchPinNotes);
    Serial.print (" ");
    Serial.print (clockPinNotes);
    Serial.print (" ");
    Serial.print (dataPinNotes);
    Serial.print (" ");
    Serial.println (LEDNotes, BIN);*/

    if (q == 0) {
        LEDNotes = 0b01000000;
        /*Serial.print ("Note is a B if LEDNote = ");
        Serial.println (LEDNote, BIN);*/
    }
    else if (q == 1) {
        LEDNotes = 0b00000001;
        /*Serial.print ("Note is a C if LEDNote = ");
        Serial.println (LEDNote, BIN);*/
    }
    else if (q == 2) {
        LEDNotes = 0b00000010;
        /*Serial.print ("Note is a D if LEDNote = ");
        Serial.println (LEDNote, BIN);*/
    }
    else if (q == 4) {
```

```

    LEDNotes = 0b00000100;
    /*Serial.print ("Note is a E if LEDNote = ");
    Serial.println (LEDNote, BIN);*/
}
else if (q == 5) {
    LEDNotes = 0b00001000;
    /*Serial.print ("Note is a F if LEDNote = ");
    Serial.println (LEDNote, BIN);*/
}
else if (q == 7) {
    LEDNotes = 0b00010000;
    /*Serial.print ("Note is a G if LEDNote = ");
    Serial.println (LEDNote, BIN);*/
}
else if (q == 8) {
    LEDNotes = 0b00100000;
    /*Serial.print ("Note is a A if LEDNote = ");
    Serial.println (LEDNote, BIN);*/
}
else {
    LEDNotes = 0b11111111;
    /*Serial.print ("If LEDNote = ");
    Serial.print (LEDNote, BIN);
    Serial.println (" no note and must be excluded ERROR");*/
}
/*Serial.print (qNote);
Serial.print (" ");
Serial.println (LEDNotes, BIN);*/
updateShiftReg(latchPinNotes, clockPinNotes, dataPinNotes, LEDNotes);

```

```

delay (100);
/*delay (500);
for (int i = 0; i < 8; i++) {
    bitSet (LEDNotes, i);
    updateShiftReg(latchPinNotes, clockPinNotes, dataPinNotes, LEDNotes);
    delay (100);
    bitWrite (LEDNotes, i, 0);
}*/
}

void updateOactiveLEDs(float j) {
    LEDOactive = 0;
    updateShiftReg(latchPinOactive, clockPinOactive, dataPinOactive, LEDOactive);
    /*Serial.print ("In 'updateToneLEDs' with latchPin, clockPin, dataPin and LEDNotes being ");
    Serial.print (latchPinNotes);
    Serial.print (" ");
    Serial.print (clockPinNotes);
    Serial.print (" ");
    Serial.print (dataPinNotes);
    Serial.print (" ");
    Serial.println (LEDNotes, BIN);*/
    Serial.print("  j in updateOactiveLEDs = ");
    Serial.println(j);
    if (j == 0) {
        LEDOactive = 0b00000001;
        /*Serial.print ("Oactive is a 0 if LEDOactive = ");
        Serial.println (LEDOactive, BIN);*/
    }
    else if (j <= 1) {

```

```
    LEDOctive = 0b00000010;
    /*Serial.print ("Octive is a 1 if LEDOctive = ");
    Serial.println (LEDOctive, BIN);*/
}
else if (j <= 2) {
    LEDOctive = 0b00000100;
    /*Serial.print ("Octive is a 2 if LEDOctive = ");
    Serial.println (LEDOctive, BIN);*/
}
else if (j <= 3) {
    LEDOctive = 0b00001000;
    /*Serial.print ("Octive is a 3 if LEDOctive = ");
    Serial.println (LEDOctive, BIN);*/
}
else if (j <= 4) {
    LEDOctive = 0b00010000;
    /*Serial.print ("Octive is a 4 if LEDOctive = ");
    Serial.println (LEDOctive, BIN);*/
}
else if (j <= 5) {
    LEDOctive = 0b00100000;
    /*Serial.print ("Octive is a 5 if LEDOctive = ");
    Serial.println (LEDOctive, BIN);*/
}
else if (j <= 6) {
    LEDOctive = 0b01000000;
    /*Serial.print ("Octive is a 6 if LEDOctive = ");
    Serial.println (LEDOctive, BIN);*/
}
```

```

else if (j <= 7) {
    LEDOactive = 0b10000000;
    /*Serial.print ("Oactive is a 7 if LEDOactive = ");
    Serial.println (LEDOactive, BIN);*/
}
else if (j <= 7.2) {
    LEDOactive = 0b11111111;
    /*Serial.print ("Oactive is a 8 if LEDOactive = ");
    Serial.println (LEDOactive, BIN);*/
}
else {
    LEDOactive = 0b01010101;
    /*Serial.print ("If LEDOactive = ");
    Serial.print (LEDOactive, BIN);
    Serial.println (" no oactive and must be excluded ERROR");*/
}

/*Serial.print (qNote);
Serial.print (" ");
Serial.println (LEDNotes, BIN);*/
updateShiftReg(latchPinOactive, clockPinOactive, dataPinOactive, LEDOactive);
delay (100);
/*delay (500);
for (int i = 0; i < 8; i++) {
    bitSet (LEDNotes, i);
    updateShiftReg(latchPinNotes, clockPinNotes, dataPinNotes, LEDNotes);
    delay (100);
    bitWrite (LEDNotes, i, 0);
}*/

```

```
}
```

```
void updateShiftReg(int latchPin, int clockPin, int dataPin, byte LEDs) {  
    digitalWrite (latchPin, LOW);  
    shiftOut(dataPin, clockPin, LSBFIRST, LEDs);  
    digitalWrite (latchPin, HIGH);  
}
```