

```
#
int ledPin = 13;
#
//led for visualization (use 13 for built-in led)
#

#
int speakerPin = 10;
#
//speaker connected to one of the PWM ports
#

#
#define c 261
#
#define d 294
#
#define e 329
#
#define f 349
#
#define g 391
#
#define gS 415
#
#define a 440
#
#define aS 455
#
#define b 466
#
#define cH 523
#
#define cSH 554
#
#define dH 587
#
#define dSH 622
#
#define eH 659
#
#define fH 698
#
#define fSH 740
#
#define gH 784
#
#define gSH 830
#
#define aH 880
#
//frequencies for the tones we're going to use
#
//used http://home.mit.bme.hu/~bako/tonecalc/tonecalc.htm to get these
#

#
void setup()
#
{
#
  pinMode(ledPin, OUTPUT);
#
  // sets the ledPin to be an output
#
  pinMode(speakerPin, OUTPUT);
#
  //sets the speakerPin to be an output
#
}
#

#
```

```

void loop()      // run over and over again
#
#
#   march();
#
#
#
void beep (unsigned char speakerPin, int frequencyInHertz, long
timeInMilliseconds)
#
#
#   digitalWrite(ledPin, HIGH);
#
#   //use led to visualize the notes being played
#
#
#   int x;
#
#   long delayAmount = (long)(750000/frequencyInHertz);
#
#   long loopTime = (long)((timeInMilliseconds*750)/(delayAmount*2));
#
#   for (x=0;x<loopTime;x++)
#
#   {
#
#       digitalWrite(speakerPin,HIGH);
#
#       delayMicroseconds(delayAmount);
#
#       digitalWrite(speakerPin,LOW);
#
#       delayMicroseconds(delayAmount);
#
#   }
#
#
#   digitalWrite(ledPin, LOW);
#
#   //set led back to low
#
#
#   delay(20);
#
#   //a little delay to make all notes sound separate
#
#
#
void march()
#
#
#
#   //for the sheet music see:
#
#   //http://www.musicnotes.com/sheetmusic/mtd.asp?ppn=MN0016254
#
#   //this is just a translation of said sheet music to frequencies / time
#   in ms
#
#   //used 750 ms for a quart note

//Hino do Corinthians

beep(speakerPin, fH, 750);

```

```
beep(speakerPin, eH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, a, 750);
beep(speakerPin, a, 750);
delay(250);
beep(speakerPin, a, 750);
beep(speakerPin, a, 750);
beep(speakerPin, b, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, b, 750);
delay(250);
beep(speakerPin, dH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, a, 750);
beep(speakerPin, gS, 750);
beep(speakerPin, gS, 750);
delay(250);
beep(speakerPin, gS, 750);
beep(speakerPin, gS, 750);
beep(speakerPin, a, 750);
beep(speakerPin, b, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, eH, 750);
delay(250);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
//beep(speakerPin, cS, 750);
beep(speakerPin, cSH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, dH, 750);
delay(250);
beep(speakerPin, dH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, fH, 750);
beep(speakerPin, aH, 750);
beep(speakerPin, fH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, cH, 750);
delay(250);
beep(speakerPin, aH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, b, 750);
delay(250);
beep(speakerPin, b, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, a, 750);
delay(250);
beep(speakerPin, fH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, a, 750);
delay(250);
beep(speakerPin, e, 750);
//beep(speakerPin, eS, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, gS, 750);
beep(speakerPin, a, 750);
beep(speakerPin, b, 750);
beep(speakerPin, cH, 750);
```

```

beep(speakerPin, dH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, fH, 750);
beep(speakerPin, fH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, eH, 750);
delay(250);
beep(speakerPin, e, 750);
//beep(speakerPin, fS, 750);
beep(speakerPin, fSH, 750);
beep(speakerPin, gS, 750);
beep(speakerPin, a, 750);
beep(speakerPin, b, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, fH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, eH, 750);
delay(250);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cSH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, dH, 750);
delay(250);
beep(speakerPin, dH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, aH, 750);
beep(speakerPin, aH, 750);
beep(speakerPin, fH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
delay(250);
beep(speakerPin, aH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, b, 750);
delay(250);
beep(speakerPin, b, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, eH, 750);
beep(speakerPin, dH, 750);
beep(speakerPin, cH, 750);
beep(speakerPin, b, 750);
beep(speakerPin, a, 750);
delay(500);

//and we're done \Ã³/
#
}

```