```
/*
  Morse Code Project

  This code will loop through a string of characters and convert these to morse code.
  It will blink two LED lights and play audio on a speaker.
 */


//**************************************************//
//   Type the String to Convert to Morse Code Here  //
//**************************************************//
char stringToMorseCode[] = " tune de nf4ac k ";

// Create variable to define the output pins
int led12 = 12;      // blink an led on output 12
int led6 = 6;        // blink an led on output 6
int audio8 = 8;      // output audio on pin 8
int note = 1200;      // music note/pitch
int tunewpm = 60;         // approximate words per minute
                  // dot length in milliseconds = 1/wpm
                  // fastdot length = 2/3 of normal dot lengh
                  // so the duty cycle becomes 33%
int wpm = 25;         // for ID or any other use
/*
  Set the speed of your morse code
  Adjust the 'dotlen' length to speed up or slow down your morse code
    (all of the other lengths are based on the dotlen)

  Here are the ratios code elements:
    Dash length = Dot length x 3
    Pause between elements = Dot length
      (pause between dots and dashes within the character)
    Pause between characters = Dot length x 3
    Pause between words = Dot length x 7

        http://www.nu-ware.com/NuCode%20Help/index.html?m...
*/
// ------Element Lengths for TUNING ----------------------------
int tunedotLen = 500/tunewpm;     // length of the morse code fast tuning 'dit' in milliseconds
int tuneelemPause = tunedotLen;  // length of the pause between elements of a character
int tuneSpaces = tunedotLen * 3;     // length of the spaces between characters

int dotLen = 1000/wpm;     // length of the morse code 'dot' in milliseconds
int dashLen = dotLen * 3;    // length of the morse code 'dash'
int elemPause = dotLen;  // length of the pause between elements of a character
int Spaces = dotLen * 3;     // length of the spaces between characters
int wordPause = dotLen * 7;  // length of the pause between words
int fastdotLen= dotLen/2 ;    // fast dot length to get 33% dutycycle
```

```
int seccounter;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output for LED lights.
  pinMode(led12, OUTPUT);
  pinMode(led6, OUTPUT);
}

// Create a loop of the letters/words you want to output in morse code (defined in string at top of code)
void loop()
{
  //send 8 seconds of fast dots
  for (seccounter=0;seccounter< (1600/(tunedotLen))  ; seccounter++)

  {
    FastDot();
    LightsOff(tunedotLen*4);
    // So the total time consumed is five * tunedotLen, giving us a 20% duty cycle here
   }
   LightsOff(1000);  // off for 1 second
   digitalWrite(led12, HIGH);    // turn the LED on
   digitalWrite(led6, HIGH);
   tone(audio8, note, 1000); // start playing a tone
   delay(1000);
   LightsOff(1000); // off for another second



  //Now send the identification

  // Loop through the string and get each character one at a time until the end is reached
  for (int i = 0; i < sizeof(stringToMorseCode) - 1; i++)
  {
    // Get the character in the current position
  char tmpChar = stringToMorseCode[i];
  // Set the case to lower case
  tmpChar = toLowerCase(tmpChar);
  // Call the subroutine to get the morse code equivalent for this character
  GetChar(tmpChar);
  LightsOff(Spaces);
  }

  // At the end of the string long pause before looping and starting again
  LightsOff(4000);
}
// FAST DOT

void FastDot()
```

```
{
  digitalWrite(led12, HIGH);   // turn LED on
  digitalWrite(led6, HIGH);
  tone(audio8,note,tunedotLen);  // play tone
  delay(tunedotLen);      // hole in this position

}
// DOT
void MorseDot()
{
  digitalWrite(led12, HIGH);    // turn the LED on
  digitalWrite(led6, HIGH);
  tone(audio8, note, dotLen); // start playing a tone
  delay(dotLen);              // hold in this position
}

// DASH
void MorseDash()
{
  digitalWrite(led12, HIGH);    // turn the LED on
  digitalWrite(led6, HIGH);
  tone(audio8, note, dashLen);  // start playing a tone
  delay(dashLen);               // hold in this position
}

// Turn Off
void LightsOff(int delayTime)
{
  digitalWrite(led12, LOW);     // turn the LED off
  digitalWrite(led6, LOW);
  noTone(audio8);             // stop playing a tone
  delay(delayTime);             // hold in this position
}

// *** Characters to Morse Code Conversion *** //
void GetChar(char tmpChar)
{
  // Take the passed character and use a switch case to find the morse code for that character
  switch (tmpChar) {
    case '1':
    MorseDot();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    MorseDash();
```

```
LightsOff(elemPause);
break;

case '2':
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;

case '3':
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;

case '4':
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;

case '5':
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
```

```
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;

case '6':
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;

case '7':
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;

case '8':
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;

case '9':
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
```

```
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;

case '0':
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;

case 'a':
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
case 'b':
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'c':
  MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'd':
MorseDash();
```

```
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'e':
MorseDot();
LightsOff(elemPause);
break;
case 'f':
  MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'g':
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'h':
 MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'i':
  MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'j':
  MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
```

```
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
  case 'k':
  MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
case 'l':
  MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
  case 'm':
  MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
case 'n':
  MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'o':
  MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
case 'p':
  MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDash();
```

```
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 'q':
  MorseDash();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
case 'r':
  MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 's':
  MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
break;
case 't':
  MorseDash();
LightsOff(elemPause);
break;
case 'u':
  MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDash();
LightsOff(elemPause);
break;
case 'v':
  MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
MorseDot();
LightsOff(elemPause);
```

```
    MorseDash();
    LightsOff(elemPause);
    break;
    case 'w':
      MorseDot();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    break;
    case 'x':
      MorseDash();
    LightsOff(elemPause);
    MorseDot();
    LightsOff(elemPause);
    MorseDot();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    break;
    case 'y':
      MorseDash();
    LightsOff(elemPause);
    MorseDot();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    break;
    case 'z':
      MorseDash();
    LightsOff(elemPause);
    MorseDash();
    LightsOff(elemPause);
    MorseDot();
    LightsOff(elemPause);
    MorseDot();
    LightsOff(elemPause);
    break;
    default:
    // If a matching character was not found it will default to a blank space
    LightsOff(Spaces);
  }
}

/*
  Unlicensed Software:
```