

## FUNCTIONS TO NOTE

**allLEDoft();**

Turns all four LEDs off.

**buttonCheck();**

Uses **digitalRead()** to check which button is pressed. Returns 0, 1, 2 or 3 if one of the buttons is pressed. Returns 4 if no button is pressed.

**startSequence();**

Flashes the LEDs and plays tones in a sequence. Resets the round counter and generates a new random sequence for the user to remember.

**winSequence();**

Plays a sequence of tones, turns all of the LEDs on, and then waits for the player to press a button. If a button is pressed, restarts the game.

**loseSequence();**

Plays a sequence of tones, turns all of the LEDs on, and then waits for the player to press a button. If a button is pressed, restarts the game.

## CODING CHALLENGES

**CHANGE THE DIFFICULTY OF THE GAME:** Change the difficulty of the game by changing how fast the player has to press each button or by increasing or decreasing the number of rounds needed to win. Note that if you increase the number of rounds to be larger than 16, you will need to change the size of the array (it is set at the top of the code in a line that looks like this: `int buttonSequence[16];` ).

**CHANGE THE SOUND EFFECTS:** Try changing the sequence of notes that plays when you start, win or lose the game.

**2-PLAYER MODE:** Try changing the code so that two players can play head-to-head.