

```
#include <Tone.h
```

```
// Gebruikt met vriendelijke toestemming van Abdullah  
Alhazmy www.Alhazmy13.net
```

```
Tone speakerpin;
```

```
int starttune[] = {NOTE_C4, NOTE_F4, NOTE_C4, NOTE_F4, NOTE_C4,  
    NOTE_F4, NOTE_C4, NOTE_F4, NOTE_G4, NOTE_F4,  
    NOTE_E4, NOTE_F4, NOTE_G4
```

```
};
```

```
int duration2[] = {100, 200, 100, 200, 100, 400, 100, 100, 100, 100,  
    200, 100, 500
```

```
};
```

```
int note[] = {NOTE_C4, NOTE_C4, NOTE_G4, NOTE_C5, NOTE_G4, NOTE_C5};
```

```
int duration[] = {100, 100, 100, 300, 100, 300};
```

```
int buttonPin[] = {2, 3, 4, 5}; // Pennen aangesloten op input van drukknoppen
```

```
int ledPin[] = {8, 9, 10, 11}; // Pennen aangesloten op ledjes
```

```
int turn = 0; // Tellen van aantal beurten
```

```
int buttonstate = 0; // Controleer status van drukknop
```

```
int randomArray[100]; // Bereik dat tot 100 inputs op kan slaan
```

```
int inputArray[100];
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    speakerpin.begin(12); // Pen aangesloten op piëzo zoemer
```

```
    for (int x = 0; x < 4; x++) {
```

```
        pinMode(ledPin[x], OUTPUT); // Stel led pennen in als output
```

```
    }
```

```
    for (int x = 0; x < 4; x++) {
```

```
        pinMode(buttonPin[x], INPUT); // Stel drukknop pennen in als inputs
```

```
        digitalWrite(buttonPin[x], HIGH); // Intern omhoogtrekken inschakelen;
```

```
        // drukknoppen starten in hoge positie
```

```
        // omgekeerde logica
```

```
    }
```

```
    // Meer "willekeur" genereren met randomArray voor de output
```

```
    // functie zodat het patroon elke keer anders is
```

```
    randomSeed(analogRead(0));
```

```
    for (int thisNote = 0; thisNote < 13; thisNote++) {
```

```
        speakerpin.play(starttune[thisNote]); // Speel de volgende noot af
```

```
        if (thisNote == 0 || thisNote == 2 || thisNote == 4 ||
```

```
            thisNote == 6) { // Houd de noot vast
```

```
            digitalWrite(ledPin[8], HIGH);
```

```
        }
```

```
        if (thisNote == 1 || thisNote == 3 || thisNote == 5 ||
```

```
            thisNote == 7 || thisNote == 9 || thisNote == 11) {
```

```
            digitalWrite(ledPin[9], HIGH);
```

```
        }
```

```
        if (thisNote == 8 || thisNote == 12) {
```

```
            digitalWrite(ledPin[10], HIGH);
```

```
        }
```

```
        if (thisNote == 10) {
```

```
            digitalWrite(ledPin[11], HIGH);
```

```
        }
```

```
        delay(duration2[thisNote]);
```

```

speakerpin.stop(); //Stop voor de volgende noot
digitalWrite(ledPin[8], LOW);
digitalWrite(ledPin[9], LOW);
digitalWrite(ledPin[10], LOW);
digitalWrite(ledPin[11], LOW);
delay(25);
}
delay(1000);
}
void loop() {
// Genereer het bereik dat de speler na moet bootsen
for (int y = 0; y <= 99; y++) {
digitalWrite(ledPin[8], HIGH);
digitalWrite(ledPin[9], HIGH);
digitalWrite(ledPin[10], HIGH);
digitalWrite(ledPin[11], HIGH);
//Speel de volgende noot af
for (int thisNote = 0; thisNote < 6; thisNote ++ ) {
speakerpin.play(note[thisNote]); // Houd de noot vast
delay(duration[thisNote]); // Stop voor de volgende noot
speakerpin.stop();
delay(25);
}
}
digitalWrite(ledPin[8], LOW);
digitalWrite(ledPin [9], LOW);
digitalWrite(ledPin[10],LOW);
digitalWrite(ledPin[11], LOW);
delay(1000);
// Beperkt door de "beurt" variabele
for (int y = turn; y <= turn; y++) {
Serial.println("");
Serial.print("Turn: ");
Serial.print(y);
Serial.println("");
randomArray[y] = random(1, 5); //Wijs een willekeurig getal toe (1-4)
// Zet ledjes in willekeurige volgorde aan
for (int x = 0; x <= turn; x++) {
Serial.print(randomArray[x]);
for (int y =0; y <4; y++) {
if (randomArray[x] == 1 && ledPin[y] == 8) {
digitalWrite(ledPin[y], HIGH);
speakerpin.play(NOTE_G3, 100);
delay(400);
digitalWrite(ledPin[y], LOW);
delay(100);
}
if (randomArray[x] == 2 && ledPin[y] == 9) {
digitalWrite(ledPin[y], HIGH);
speakerpin.play(NOTE_A3, 100);
delay(400);
digitalWrite(ledPin[y], LOW);
delay(100);
}
if (randomArray[x] == 3 && ledPin[y] == 10) {
digitalWrite(ledPin[y], HIGH); speakerpin.play(NOTE_B3, 100);
}
}
}
}

```

```

delay(400); digitalWrite(ledPin[y], LOW);
delay(100);
}
if (randomArray[x] == 4 && ledPin[y] == 11) {
  digitalWrite(ledPin[y], HIGH);
  speakerpin.play(NOTE_C4, 100);
  delay(400);
  digitalWrite(ledPin[y], LOW);
  delay(100);
}
}
}
}
input();
}
}

```

// Controleer of de input overeenkomt met het patroon

```

void input() {
  for (int x = 0; x <= turn;) {
    for (int y = 0; y < 4; y++) {
      buttonstate = digitalRead(buttonPin[y]); // Controleer of knop wordt ingedrukt
      if (buttonstate == LOW && buttonPin[y] == 2) {
        digitalWrite(ledPin[8], HIGH);
        speakerpin.play(NOTE_G3, 100);
        delay(200);
        digitalWrite(ledPin[8], LOW);
        inputArray[x] = 1;
        delay(250);
        Serial.print(" ");
        Serial.print(1);
        //Controleer of input van gebruiker overeenkomt met gegenereerd patroon
        if (inputArray[x] != randomArray[x]) {
          fail(); // Zo niet, dan wordt functie "mislukt" aangeroepen
        }
        x++;
      }
      if (buttonstate == LOW && buttonPin[y] == 3) {
        digitalWrite(ledPin[9], HIGH);
        speakerpin.play(NOTE_A3, 100);
        delay(200);
        digitalWrite(ledPin[9], LOW);
        inputArray[x] = 2;
        delay(250);
        Serial.print(" ");
        Serial.print(2);
        if (inputArray[x] != randomArray[x]) {
          fail();
        }
      }
    }
  }
}

```

```

    x++;
}
if (buttonstate == LOW && buttonPin[y] == 4) {
    digitalWrite(ledPin[10], HIGH);
    speakerpin.play(NOTE_B3, 100);
    delay(200);
    digitalWrite(ledPin[10], LOW);
    inputArray[x] = 3;
    delay(250);
    Serial.print(" ");
    Serial.print(3);
}
if (inputArray[x] != randomArray[x]) {
    fail();
}
x++;
}
if (buttonstate == LOW && buttonPin[y] == 5) {
    digitalWrite(ledPin[11], HIGH);
    speakerpin.play(NOTE_C4, 100);
    delay(200);
    digitalWrite(ledPin[11], LOW);
    inputArray[x] = 4;
    delay(250);
    Serial.print(" ");
    Serial.print(4);
}
if (inputArray[x] != randomArray[x]) {
    fail();
}
x++;
}
}
delay(500);
turn++; // Verhoog aantal beurten
}
// Functie die gebruikt wordt als speler er niet in slaagt volgorde na te bootsen
void fail() {
    for (int y = 0; y <= 2; y++) { // Knipper met lampjes om aan te geven dat het mislukt is
        digitalWrite(ledPin[8], HIGH);
        digitalWrite(ledPin[9], HIGH);
        digitalWrite(ledPin[10], HIGH);
        digitalWrite(ledPin[11], HIGH);
        speakerpin.play(NOTE_G3, 300);
        delay(200);
        digitalWrite(ledPin[8], LOW);
        digitalWrite(ledPin[9], LOW);
        digitalWrite(ledPin[10], LOW);
        digitalWrite(ledPin[11], LOW);
    }
}

```

```
speakerpin.play(NOTE_C3, 300);  
delay(200);  
}  
delay(500);  
turn = -1; // Reset waarde "beurten" om spel opnieuw te starten  
}  
[/code]
```