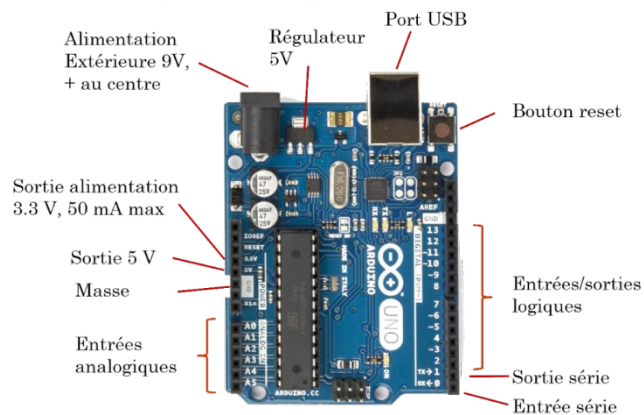


## La boîte à mystère

Ce projet consiste à créer une boîte où il faudrait trouver un certain nombre de réponses avec des inscriptions et des défis sur certaines faces du cube. Si vous trouvez toutes les bonnes réponses, une musique se déclenche et dit que vous avez gagné. Sinon une autre musique se déclenche pour vous dire que vous avez perdu !

### Présentation de la carte ARDUINO UNO



### Liste du matériel utilisé :

- un buzzer
- une bande de led
- un Lecteur LCD
- un bouton poussoir
- une carte Arduino UNO
- plusieurs câbles

### Fonctionnement des actionneurs utilisés :

#### Buzzer :



Le buzzer est ici utilisé dans deux circonstances :

- la première est celle quand on gagne le jeu, une certaine musique se déclenche grâce aux instructions suivante :  
`afficheur_d () ;`

```

if (digitalRead(pin_button) == HIGH) { // si le bouton est pressé ...
  q_emoji ();
  delay (3000);
  // FIN V
if (readValue < 514) {
  gagne() ;
  af_gagne();
  buzzer.begin(100);

```

-ensuite, la deuxième circonstance est celle quand on perd au jeu, une autre musique ce déclenche grâce aux instructions suivante :

```

else {
  af_perdu ();
  perdu() ;
  buzzer.begin(10);

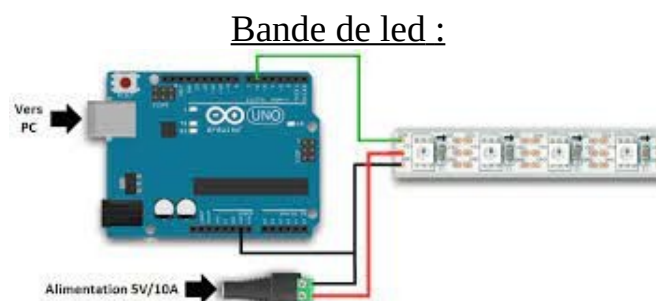
```

Dans les deux cas, ce qui précèdera ces instructions sont des notes de musique que le buzzer jouera, comme les suivantes :

```

buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_F3, 375);
buzzer.sound(NOTE_C4, 125);

```



La bande led dispose de 30 leds. La couleur de chaque led peut être contrôlée soit individuellement, soit toute en même temps avec trois valeurs correspondantes à 3 couleurs qui sont le rouge le vert et le bleu. Si l'on veut avoir par exemple du violet, il faudra modifier l'intensité de celle-ci.

En début de programme, on crée une occurrence et on lui donne les paramètres que l'on veut (le port où le ruban est connecté et le nombre de leds qui doivent s'allumer).

```

#define PIN      6
#define NUMPIXELS 30

```

Pour dire une couleur à toute les leds, on doit créer un tuple avec en première information rouge, vert, puis bleu.

```

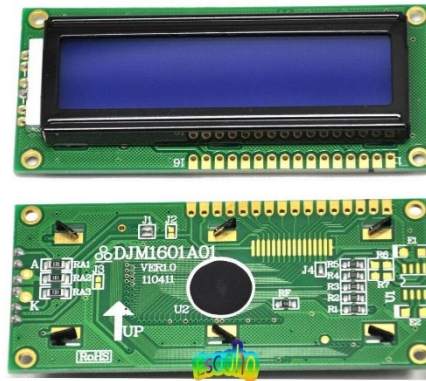
Pixels.setPixelColor(i, pixels.Color(250, 0, 0)) ;

```

Ces valeurs doivent donc être comprises entre 0 et 255 pour que les leds affichent une certaine couleur.

Dans notre programme, la bande led affichera soit du vert quand le joueur gagne et sinon elle devienne rouge. En début de jeu celle-ci s'allumeront bleu.

### Lecteur LCD :



Le lecteur LCD fait parti des pièces les plus importantes de notre projet, elle affichera toutes les questions à poser à l'utilisateur mais aussi affichera s'il a gagné ou s'il a perdu et donc te donnera toutes les instructions nécessaires pour que le jeu se déroule. Celui-ci est défini par l'instruction :

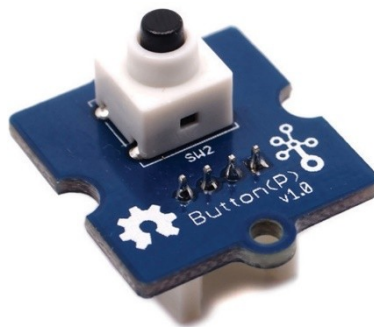
```
#include <Wire.h>
```

```
#include "rgb_lcd.h"
```

Il va être ensuite définis avec 16 colonnes et 2 lignes qui feront donc 32 caractères :

```
lcd.begin(16, 2);
```

### Bouton poussoir :



Pour finir, nous avons besoin d'un bouton poussoir pour répondre au question.

Celui-ci est définis par l'instructions :

```
pinMode(pin_button, INPUT)
```

Pour que l'utilisateur puisse répondre à chaque question, cette instructions sera répété pour chaque question posé au joueur

## Code python du projet :

```
#include <Adafruit_LiquidCrystal.h>

// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// Released under the GPLv3 license to match the rest of the
// Adafruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN      6 // On Trinket or Gemma, suggest changing this to 1

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS 30 // Popular NeoPixel ring size

// When setting up the NeoPixel library, we tell it how many pixels,
// and which pin to use to send signals. Note that for older NeoPixel
// strips you might need to change the third parameter -- see the
// strandtest example for more information on possible values.
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

#define DELAYVAL 100 // Time (in milliseconds) to pause between pixels
int potPin = A0; // Le potentiomètre est connecté à la broche analogue 0
int readValue;

// LECTEUR LCD
#include <Wire.h>
#include "rgb_lcd.h"

rgb_lcd lcd;

#include <Buzzer.h>

Buzzer buzzer(5);

int pin_button = 2; // port numérique lié au bouton poussoir

// SIGNE POUR QUIZ
// make some custom characters:
byte heart[8] = {
  0b00000,
```

```
    0b01010,  
    0b11111,  
    0b11111,  
    0b11111,  
    0b01110,  
    0b00100,  
    0b00000  
};
```

```
byte smiley[8] = {  
    0b00000,  
    0b00000,  
    0b01010,  
    0b00000,  
    0b00000,  
    0b10001,  
    0b01110,  
    0b00000  
};
```

```
byte frownie[8] = {  
    0b00000,  
    0b00000,  
    0b01010,  
    0b00000,  
    0b00000,  
    0b00000,  
    0b01110,  
    0b10001  
};
```

```
byte armsDown[8] = {  
    0b00100,  
    0b01010,  
    0b00100,  
    0b00100,  
    0b01110,  
    0b10101,  
    0b00100,  
    0b01010  
};
```

```
byte armsUp[8] = {  
    0b00100,  
    0b01010,
```

```
    0b00100,  
    0b10101,  
    0b01110,  
    0b00100,  
    0b00100,  
    0b01010  
};  
byte gun[8] = {  
    0b01000,  
    0b11000,  
    0b01000,  
    0b01000,  
    0b01100,  
    0b11010,  
    0b01111,  
    0b00000  
};  
int tab[] = {0, 1, 2, 3, 4, 5, 6};
```

```
void setup() {  
    // These lines are specifically to support the Adafruit Trinket 5V 16 MHz.  
    // Any other board, you can remove this part (but no harm leaving it):  
#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)  
    clock_prescale_set(clock_div_1);  
#endif  
    // END of Trinket-specific code.  
  
    pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)  
  
    /* Initialise le port série pour le debug */  
    Serial.begin(9600);  
  
    /* Initialise le générateur de nombre aléatoire avec une graine aléatoire */  
    randomSeed(analogRead(0));  
  
    // afficheur lcd  
    // set up the LCD's number of columns and rows:  
    lcd.begin(16, 2);
```

```

lcd.setRGB(255, 0, 250 );

// Print a message to the LCD.

  lcd.clear();

// bouton
pinMode(pin_button, INPUT); // réglage du port du bouton en mode ENTREE

// PERSO QUIZ
// create a new character
lcd.createChar(6, heart);
// create a new character
lcd.createChar(1, smiley);
// create a new character
lcd.createChar(2, frownie);
// create a new character
lcd.createChar(3, armsDown);
// create a new character
lcd.createChar(4, armsUp);
lcd.createChar(5, gun);

}

void gagne() {
  pixels.clear(); // Set all pixel colors to 'off'

  // The first NeoPixel in a strand is #0, second is 1, all the way up
  // to the count of pixels minus one.
  for(int i=0; i<NUMPIXELS; i++) { // For each pixel...

    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(0, 250, 0));

    pixels.show(); // Send the updated pixel colors to the hardware.

    delay(DELAYVAL); // Pause before next pass through loop
  }
}

void perdu () {
  pixels.clear(); // Set all pixel colors to 'off'

```

```

// The first NeoPixel in a strand is #0, second is 1, all the way up
// to the count of pixels minus one.
for(int i=0; i<NUMPIXELS; i++) { // For each pixel...

    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(250, 0, 0));
    delay(10);

    pixels.show(); // Send the updated pixel colors to the hardware.
}
pixels.clear(); // Set all pixel colors to 'off'

```

```

// The first NeoPixel in a strand is #0, second is 1, all the way up
// to the count of pixels minus one.
for(int i=0; i<NUMPIXELS; i++) { // For each pixel...
    delay (20) ;
    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(250, 0, 0));

    pixels.show(); // Send the updated pixel colors to the hardware.
}
pixels.clear(); // Set all pixel colors to 'off'

```

```

// The first NeoPixel in a strand is #0, second is 1, all the way up
// to the count of pixels minus one.
for(int i=0; i<NUMPIXELS; i++) { // For each pixel...
    delay (30) ;
    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(250, 0, 0));

    pixels.show(); // Send the updated pixel colors to the hardware.
}
}

```

```

void afficheur_d () {
    // set up the LCD's number of columns and rows:
    pixels.clear(); // Set all pixel colors to 'off'
}

```

```

// The first NeoPixel in a strand is #0, second is 1, all the way up
// to the count of pixels minus one.
for(int i=0; i<NUMPIXELS; i++) {// For each pixel...

    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(0, 150, 250));
    delay(10);

    pixels.show(); // Send the updated pixel colors to the hardware.
}

lcd.begin(16, 2);

lcd.setRGB(0, 200, 250 );

// Print a message to the LCD.
lcd.print("Play ?");
delay(2000) ;

lcd.setCursor(4, 1);
lcd.setRGB(0, 200, 250 );
lcd.print("press button");
delay(2000);

    lcd.clear();
}

void af_perdu() { // GAGNER
    lcd.setRGB(250, 50, 0 );
    lcd.print("DOMMAGE PERDU!");
    delay(5000);

}

void af_gagne () { // GAGNER
    lcd.setRGB(0, 50, 250 );
    lcd.print("Bravo,gagné !");
    delay(5000);

}

```

```

}
void loop() {

  /* Génère un nombre aléatoire entre 0 et 5 et l'affiche sur le port série */

  /*Inscription dans le moniteur série
  de "Color=valeur de la variable readValue
  */

  delay(100);

  // De 0 à 514, GAGNE
  afficheur_d ();
  if (digitalRead(pin_button) == HIGH) { // si le bouton est pressé ...
    lcd.clear() ;
    lcd.print("1 question!");
    delay(2000) ;

    lcd.setCursor(4, 1);
    lcd.setRGB(0, 200, 250 );
    lcd.print("Attention !");
    delay(2000);
    lcd.clear() ;

    // reponse toute première Arduino en 2005
    lcd.setCursor(0, 0);
    lcd.print("annee creation");
    lcd.setCursor(0, 1);
    lcd.print("carte arduino ?");
    delay(2000);
    lcd.clear() ;
    lcd.setCursor(0, 0);
    lcd.print("1, en 1998");
    delay(5000);
    if (digitalRead(pin_button) == HIGH) {
      lcd.clear() ;
      af_perdu ();
      perdu() ;
      buzzer.begin(10);

      buzzer.sound(NOTE_A3, 500);
    }
  }
}

```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);
```

```
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();  
}
```

```
lcd.clear() ;  
lcd.print("2, en 2000");  
delay(5000);
```

```
if (digitalRead(pin_button) == HIGH) {  
  lcd.clear() ;  
  af_perdu ();  
  perdu() ;  
  buzzer.begin(10);
```

```
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_F3, 375);  
  buzzer.sound(NOTE_C4, 125);
```

```
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_F3, 375);  
  buzzer.sound(NOTE_C4, 125);  
  buzzer.sound(NOTE_A3, 1000);
```

```
  buzzer.sound(NOTE_E4, 500);  
  buzzer.sound(NOTE_E4, 500);  
  buzzer.sound(NOTE_E4, 500);  
  buzzer.sound(NOTE_F4, 375);  
  buzzer.sound(NOTE_C4, 125);
```

```
  buzzer.sound(NOTE_GS3, 500);  
  buzzer.sound(NOTE_F3, 375);  
  buzzer.sound(NOTE_C4, 125);  
  buzzer.sound(NOTE_A3, 1000);
```

```
  buzzer.sound(NOTE_A4, 500);  
  buzzer.sound(NOTE_A3, 375);  
  buzzer.sound(NOTE_A3, 125);  
  buzzer.sound(NOTE_A4, 500);  
  buzzer.sound(NOTE_GS4, 375);  
  buzzer.sound(NOTE_G4, 125);
```

```
  buzzer.sound(NOTE_FS4, 125);  
  buzzer.sound(NOTE_E4, 125);  
  buzzer.sound(NOTE_F4, 250);  
  buzzer.sound(0, 250);  
  buzzer.sound(NOTE_AS3, 250);  
  buzzer.sound(NOTE_DS4, 500);  
  buzzer.sound(NOTE_D4, 375);  
  buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();  
}
```

```
lcd.clear() ;  
lcd.print("3,en 2005");  
delay(5000);  
if (digitalRead(pin_button) == HIGH) {  
  lcd.clear() ;  
  // PASSAGE QUESTION 2  
  lcd.print("2 question!");  
delay(2000) ;
```

```
  lcd.setCursor(4, 1);  
  lcd.setRGB(0, 200, 250 );  
  lcd.print("Attention !");  
  delay(2000);  
lcd.clear() ;
```

```
  // reponse toute première Arduino en 2005  
lcd.setCursor(0, 0);  
lcd.print("Qu est ce un ");  
lcd.setCursor(0, 1);  
lcd.print("lecteur LDC ?");  
  delay(2000);  
lcd.clear() ;  
lcd.setCursor(0, 0);  
lcd.print("1, un bouton");  
delay(5000);  
if (digitalRead(pin_button) == HIGH) {  
  lcd.clear() ;  
  af_perdu ();  
  perdu() ;  
  buzzer.begin(10);
```

```
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_F3, 375);  
  buzzer.sound(NOTE_C4, 125);
```

```
  buzzer.sound(NOTE_A3, 500);  
  buzzer.sound(NOTE_F3, 375);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);
```

```
buzzer.sound(NOTE_A3, 375);
buzzer.sound(NOTE_A3, 125);
buzzer.sound(NOTE_A4, 500);
buzzer.sound(NOTE_GS4, 375);
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);
buzzer.sound(NOTE_E4, 125);
buzzer.sound(NOTE_F4, 250);
buzzer.sound(0, 250);
buzzer.sound(NOTE_AS3, 250);
buzzer.sound(NOTE_DS4, 500);
buzzer.sound(NOTE_D4, 375);
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);
buzzer.sound(NOTE_B3, 125);
buzzer.sound(NOTE_C4, 250);
buzzer.sound(0, 250);
buzzer.sound(NOTE_F3, 250);
buzzer.sound(NOTE_GS3, 500);
buzzer.sound(NOTE_F3, 375);
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_F3, 375);
buzzer.sound(NOTE_C4, 125);
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);
  lcd.clear();
}
lcd.clear() ;
lcd.print("2,lecteur disque");
delay(5000);
if (digitalRead(pin_button) == HIGH) {
  lcd.clear() ;
  af_perdu ();
  perdu() ;
  buzzer.begin(10);
```

```
buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_A3, 500);
buzzer.sound(NOTE_F3, 375);
```

```
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();  
}  
lcd.clear() ;  
lcd.print("3,un ustensil");  
delay(5000);  
  if (digitalRead(pin_button) == HIGH) {  
    lcd.clear() ;  
    af_perdu ();  
    perdu() ;
```

```
buzzer.begin(10);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);
```

```
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();  
}  
lcd.clear() ;
```

```

lcd.print("4,un ecran");
delay(5000);
if (digitalRead(pin_button) == HIGH) {
  lcd.clear() ;

  lcd.clear() ;

  //question 3

  lcd.print("3 question!");
  delay(2000) ;

  lcd.setCursor(4, 1);
  lcd.setRGB(0, 200, 250 );
  lcd.print("Attention !");
  delay(2000);
  lcd.clear() ;

  // reponse qui est le createur d'ardueau
  lcd.setCursor(0, 0);
  lcd.print("qui est le");
  lcd.setCursor(0, 1);
  lcd.print("createur arduino");
  delay(2000);
  lcd.clear() ;
  lcd.setCursor(0, 0);
  lcd.print("1,Antoine dupont");
  delay(5000);
  if (digitalRead(pin_button) == HIGH) {
    lcd.clear() ;
    af_perdu ();
    perdu() ;
    buzzer.begin(10);

    buzzer.sound(NOTE_A3, 500);
    buzzer.sound(NOTE_A3, 500);
    buzzer.sound(NOTE_A3, 500);
    buzzer.sound(NOTE_F3, 375);
    buzzer.sound(NOTE_C4, 125);

    buzzer.sound(NOTE_A3, 500);

```

```
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();  
}
```

```
lcd.clear() ;  
lcd.print("2, Tom Malaurie");  
delay(5000);  
if (digitalRead(pin_button) == HIGH) {  
  lcd.clear() ;  
  af_perdu ();  
  perdu() ;  
  buzzer.begin(10);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);
```

```
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();  
  lcd.clear() ;  
}
```

```
lcd.clear() ;  
lcd.print("3Nathanael Dumas");  
delay(5000);  
  if (digitalRead(pin_button) == HIGH) {
```

```
lcd.clear() ;  
af_perdu ();  
perdu() ;  
buzzer.begin(10);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);
```

```
    lcd.clear();
}
lcd.clear() ;
lcd.print("4, Tom Igoe");
delay(5000);
if (digitalRead(pin_button) == HIGH) {
    lcd.clear() ;
    gagne() ;
    af_gagne();
    buzzer.begin(100);
```

```
    buzzer.sound(NOTE_E7, 80);
    buzzer.sound(NOTE_E7, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_E7, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_C7, 80);
    buzzer.sound(NOTE_E7, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_G7, 80);
    buzzer.sound(0, 240);
    buzzer.sound(NOTE_G6, 80);
    buzzer.sound(0, 240);
    buzzer.sound(NOTE_C7, 80);
    buzzer.sound(0, 160);
    buzzer.sound(NOTE_G6, 80);
    buzzer.sound(0, 160);
    buzzer.sound(NOTE_E6, 80);
    buzzer.sound(0, 160);
    buzzer.sound(NOTE_A6, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_B6, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_AS6, 80);
    buzzer.sound(NOTE_A6, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_G6, 100);
    buzzer.sound(NOTE_E7, 100);
    buzzer.sound(NOTE_G7, 100);
    buzzer.sound(NOTE_A7, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_F7, 80);
    buzzer.sound(NOTE_G7, 80);
    buzzer.sound(0, 80);
    buzzer.sound(NOTE_E7, 80);
```

```
buzzer.sound(0, 80);
buzzer.sound(NOTE_C7, 80);
buzzer.sound(NOTE_D7, 80);
buzzer.sound(NOTE_B6, 80);
buzzer.sound(0, 160);
buzzer.sound(NOTE_C7, 80);
buzzer.sound(0, 160);
buzzer.sound(NOTE_G6, 80);
buzzer.sound(0, 160);
buzzer.sound(NOTE_E6, 80);
buzzer.sound(0, 160);
buzzer.sound(NOTE_A6, 80);
buzzer.sound(0, 80);
buzzer.sound(NOTE_B6, 80);
buzzer.sound(0, 80);
buzzer.sound(NOTE_AS6, 80);
buzzer.sound(NOTE_A6, 80);
buzzer.sound(0, 80);
buzzer.sound(NOTE_G6, 100);
buzzer.sound(NOTE_E7, 100);
buzzer.sound(NOTE_G7, 100);
buzzer.sound(NOTE_A7, 80);
buzzer.sound(0, 80);
buzzer.sound(NOTE_F7, 80);
buzzer.sound(NOTE_G7, 80);
buzzer.sound(0, 80);
buzzer.sound(NOTE_E7, 80);
buzzer.sound(0, 80);
buzzer.sound(NOTE_C7, 80);
buzzer.sound(NOTE_D7, 80);
buzzer.sound(NOTE_B6, 80);
buzzer.sound(0, 160);
```

```
buzzer.end(2000);
  lcd.clear();
```

```
}
}
}
}
```

```
lcd.clear() ;
lcd.print("4, en 2003");
delay(5000);
if (digitalRead(pin_button) == HIGH) {
  lcd.clear() ;
```

```
af_perdu ();  
perdu() ;  
buzzer.begin(10);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_E4, 500);  
buzzer.sound(NOTE_F4, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_A3, 125);
```

```
buzzer.sound(NOTE_C4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_E4, 1000);
```

```
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_A3, 375);  
buzzer.sound(NOTE_A3, 125);  
buzzer.sound(NOTE_A4, 500);  
buzzer.sound(NOTE_GS4, 375);  
buzzer.sound(NOTE_G4, 125);
```

```
buzzer.sound(NOTE_FS4, 125);  
buzzer.sound(NOTE_E4, 125);  
buzzer.sound(NOTE_F4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_AS3, 250);  
buzzer.sound(NOTE_DS4, 500);  
buzzer.sound(NOTE_D4, 375);  
buzzer.sound(NOTE_CS4, 125);
```

```
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_B3, 125);  
buzzer.sound(NOTE_C4, 250);  
buzzer.sound(0, 250);  
buzzer.sound(NOTE_F3, 250);  
buzzer.sound(NOTE_GS3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);
```

```
buzzer.sound(NOTE_A3, 500);  
buzzer.sound(NOTE_F3, 375);  
buzzer.sound(NOTE_C4, 125);  
buzzer.sound(NOTE_A3, 1000);
```

```
buzzer.end(2000);  
  lcd.clear();
```

```
}  
lcd.clear() ;
```

```
// FIN V
```

```
}
```

#### Extension ou développement possibles :

On pourrait rajouter d'autres fonctions comme :

-dire si la réponse choisie est plus ou moins loin de la bonne réponse.

-faire un compte a rebours avec la bande led pour trouver la réponse.

On pourrait également rajouter un capteur à ultrasons qui dès que le capteur détecte un son, le jeu commence.

Enfin ce dispositif pourrait être intégré dans une boite fermée pour un jeu dont le but est de trouvé des indices, de répondre correctement aux réponses et de gagner la récompense dans la boîte.