DAY 1)

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()

{

  lcd.init();         // initialize the lcd

  lcd.backlight();    // Turn on the LCD screen backlight

}

void loop()

{

  lcd.setCursor(1, 0);

  lcd.print("HI BUDDIES!");

  delay(3000);

  lcd.clear();

  lcd.setCursor(2, 0);

  lcd.print("Chunks Here");

  lcd.setCursor(2, 1);

  lcd.print("1st Display");

  delay(3000);

  lcd.clear();

  lcd.setCursor(2, 0);

  lcd.print(" WE Made");

  lcd.setCursor(3, 1);

  lcd.print(" IT ");

  delay(3000);

  lcd.clear();

}

DAY2)saltring error max sensors

DAY3)pulse

/\*  Getting\_BPM\_to\_Monitor prints the BPM to the Serial Monitor, using the least lines of code and PulseSensor Library.

 \*  Tutorial Webpage: https://pulsesensor.com/pages/getting-advanced

 \*

--------Use This Sketch To------------------------------------------

1) Displays user's live and changing BPM, Beats Per Minute, in Arduino's native Serial Monitor.

2) Print: "♥  A HeartBeat Happened !" when a beat is detected, live.

2) Learn about using a PulseSensor Library "Object".

4) Blinks the builtin LED with user's Heartbeat.

--------------------------------------------------------------------\*/

#define USE\_ARDUINO\_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.

#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.

//  Variables

const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0

const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.

int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.

                               // Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.

                               // Otherwise leave the default "550" value.

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called "pulseSensor"

void setup() {

  Serial.begin(9600);          // For Serial Monitor

  // Configure the PulseSensor object, by assigning our variables to it.

  pulseSensor.analogInput(PulseWire);

  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.

  pulseSensor.setThreshold(Threshold);

  // Double-check the "pulseSensor" object was created and "began" seeing a signal.

   if (pulseSensor.begin()) {

    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or on Arduino reset.

  }

}

void loop() {

if (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if "a beat happened".

int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".

                                               // "myBPM" hold this BPM value now.

 Serial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".

 Serial.print("BPM: ");                        // Print phrase "BPM: "

 Serial.println(myBPM);                        // Print the value inside of myBPM.

}

  delay(20);                    // considered best practice in a simple sketch.

}

Day4 sound sensor

const int ledpin=13; // ledpin and soundpin are not changed throughout the process

const int soundpin=A2;

const int threshold=500; // sets threshold value for sound sensor

void setup() {

Serial.begin(9600);

pinMode(ledpin,OUTPUT);

pinMode(soundpin,INPUT);

}

void loop() {

int soundsens=analogRead(soundpin); // reads analog data from sound sensor

if (soundsens>=threshold) {

digitalWrite(ledpin,HIGH); //turns led on

delay(10);

}

else{

digitalWrite(ledpin,LOW);

}

}