```
#Maquette SAISONS
#include <LiquidCrystal.h>

int tempPin1 = 0;       // schéma
int tempPin2 = 1;
int relay1 = 22;
int relay2 = 26;
int buttonApin = 24;

//              BS  E  D4 D5  D6 D7       // LCD1602
LiquidCrystal lcd1(8, 9, 10, 11, 12, 13);
LiquidCrystal lcd2(2, 3, 4, 5, 6, 7);

int tempReading1;     // variables
int tempReading2;
double tempK1;
double tempK2;
float tempC1;
float tempC2;
float totC1;
float totC2;
float mtempC1;
float mtempC2;
int duree;
float ecart;
float delta;
int i;
float maxecart;
unsigned long previousMillis;


void setup()  {
  lcd1.begin(16, 2);
  lcd2.begin(16, 2);
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(buttonApin, INPUT_PULLUP);
}

void loop() {

  digitalWrite(relay2, LOW);       // hello !
```

```
lcd1.clear();
lcd1.print("PRESSER BOUTON");
lcd1.setCursor(0, 1);
lcd1.print("POUR DEMARRER");
lcd2.clear();
lcd2.print("COMPARAISON FLUX");
lcd2.setCursor(0, 1);
lcd2.print("ETE / HIVER");

while ( digitalRead(buttonApin) == HIGH)
{
  delay(10);          // en attente du bouton SOLEIL
}

lcd1.clear();       // go !
lcd2.clear();
lcd1.print("DEBUT DE ");
lcd1.setCursor(0, 1);
lcd1.print("L'EXPERIENCE....");
lcd2.print("  DUREE 45 s ");
lcd2.setCursor(0, 1);
lcd2.print("     ....");


digitalWrite(relay2, LOW);
delay(3000);    // le relay2 éteint le ventilateur

digitalWrite(relay1, HIGH);    // le relay1 allume les lampes

previousMillis = millis();  // démarrage du chrono pour l'exp.

ecart = 0;
maxecart = 0;

while (millis() - previousMillis <= 45000) {
            // la mesure va se faire toutes les 100 ms pendant 45000 ms
  totC1 = 0;
  totC2 = 0;

  for (i = 1; i <= 10; i++) {
            // calcul de la température : moyenne sur 10 mesures
    tempReading1 = analogRead(tempPin1);
    tempReading2 = analogRead(tempPin2);
    tempK1 = log(10000.0 * ((1024.0 / tempReading1 - 1)));
```

```
    tempK1 = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 *
tempK1 * tempK1 )) * tempK1 );
            //  Temp1 Kelvin
    tempC1 = tempK1 - 273.15;          // Conversion Kelvin en Celcius
    tempK2 = log(10000.0 * ((1024.0 / tempReading2 - 1)));
    tempK2 = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 *
tempK2 * tempK2 )) * tempK2 );
            //  Temp2 Kelvin
    tempC2 = tempK2 - 273.15;          // Conversion Kelvin en Celcius
    totC1 = totC1 + tempC1;
    totC2 = totC2 + tempC2;
    delay(100);
  }

  duree = round((millis() - previousMillis) / 1000);
  mtempC1 = totC1 / 10.0;
  mtempC2 = totC2 / 10.0;
  ecart = mtempC2 - mtempC1;
  if (maxecart < ecart) maxecart = ecart;


  lcd2.clear();                // Display Temperature en °C
  lcd2.setCursor(0, 0);
  lcd2.print("T");
  lcd2.print((char)223);
  lcd2.print(" HIVER");
  lcd2.setCursor(9, 0);
  lcd2.print(mtempC1, 1);
  lcd2.print((char)223);
  lcd2.print("C");
  lcd2.setCursor(0, 1);
  lcd2.print("...IL RESTE ");
  lcd2.print(45 - duree);
  lcd2.print(" S");

  lcd1.clear();
  lcd1.setCursor(0, 0);
  lcd1.print("T");
  lcd1.print((char)223);
  lcd1.print(" ETE");
  lcd1.setCursor(9, 0);
  lcd1.print(mtempC1 + ecart, 1);
  lcd1.print((char)223);
  lcd1.print("C");
```

```
    lcd1.setCursor(0, 1);
    lcd1.print("EXP. EN COURS...");

  }

  digitalWrite(relay1, LOW);               // stop les spots
  lcd1.clear();
  lcd2.clear();
  lcd2.print("FIN DE LA MESURE");         // résultat final
  lcd1.print("ECART        C");
  lcd1.setCursor(12, 0);
  lcd1.print((char)223);
  lcd1.setCursor(8, 0);
  lcd1.print(maxecart, 1);
  delay(2000);

  digitalWrite(relay2, HIGH);              // ventilation
  delay(2000);
  delta = 11;         // assure au moins 3s de ventilation
  lcd1.clear();
  lcd2.clear();
  previousMillis = millis();

  while (delta > 7 )   {
 // la ventilation se fera tant que les température des sondes sont trop
différentes >0,5°C

    delta = abs(analogRead(tempPin1) - analogRead(tempPin2));
    lcd2.print("VENTILATION ...");
    delay(1500);
    lcd2.clear();
    lcd1.print("PATIENTEZ ...");
    delay(1500);
    lcd1.clear();
    if (millis() - previousMillis > 30000) break;
              // termine la ventilation en cas de problème. limite à 30s.
  }

  digitalWrite(relay2, LOW);

}
```