```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 13;

// defines variables
long duration;
int distance;
int safetyDistance;


void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);
lcd.init();
  lcd.backlight();
  lcd.print ("NO Obstacle");

delay(2000);
lcd.clear(); // Starts the serial communication
}


void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

```
// Reads the echoPin, returns the sound wave travel time in
microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= (duration*0.0347/2);

safetyDistance = distance;
if (safetyDistance <= 100){
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, LOW);
lcd.clear();
}
else{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, HIGH);
  distance = 0;
  if(distance == 0){
    lcd.setCursor(0,0);
    lcd.print("NO OBSTACLE");
    delay(500);
    lcd.clear();


  }

}

// Prints the distance on the Serial Monitor
if(distance > 0){
  lcd.setCursor(0,0);
lcd.print("OBJECT DETECTED");
lcd.setCursor(0,1);
  lcd.print(distance);
lcd.setCursor(3,1);
    lcd.print("cms");
    delay(1000);
  lcd.clear();
}




}
```