

on start

configure LCD 1 type to Standard with pins: RS 8 E 9 DB4 10 DB5 11 DB6 12 DB7 13

set position on LCD 1 to column 0 row 0

print to LCD 1 WELCOME

wait 3 secs

on LCD 1 clear the screen

forever

comment READ DISTANCE FROM ULTRASONIC SENSORS

set distSensor1 to read ultrasonic distance sensor on trigger pin A3 echo pin A2 in units cm

set distSensor2 to read ultrasonic distance sensor on trigger pin A1 echo pin A0 in units cm

comment DETECT INCOMING

if distSensor1 < 50 then

set flag1 to 1

if flag2 = 0 then

set person to person + 1

wait 1 secs

comment DETECT OUTGOING

if distSensor2 < 50 then

set flag2 to 1

if flag1 = 0 then

if person ≥ 1 then

set person to person - 1

wait 1 secs

comment SET FLAGS TO ZERO

if distSensor1 > 50 and distSensor2 > 50 and flag1 = 1 and

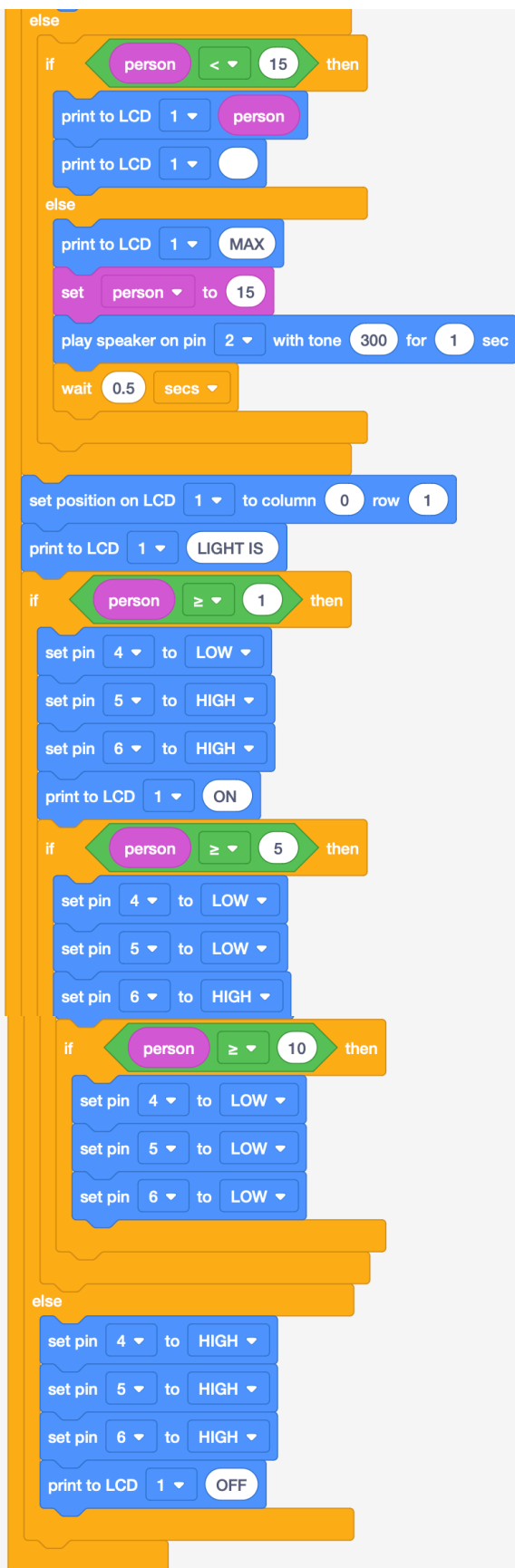
set flag1 to 0

set flag2 to 0

comment DISPLAY IN LCD SCREEN

set position on LCD 1 to column 0 row 0

print to LCD 1 HAVE PERSON



CODE:-

```
// C++ code
//
#include <LiquidCrystal.h>

int person = 0;

int distSensor1 = 0;

int distSensor2 = 0;

int flag1 = 0;

int flag2 = 0;

int LDRvalue = 0;

LiquidCrystal lcd_1(8, 9, 10, 11, 12, 13);

long readUltrasonicDistance(int triggerPin, int echoPin)
{
    pinMode(triggerPin, OUTPUT); // Clear the trigger
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Sets the trigger pin to HIGH state for 10 microseconds
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
```

```
    // Reads the echo pin, and returns the sound wave travel
    time in microseconds
    return pulseIn(echoPin, HIGH);
}
```

```
void setup()
{
    lcd_1.begin(16, 2); // Set up the number of columns and
    rows on the LCD.
    pinMode(2, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);

    lcd_1.setCursor(0, 0);
    lcd_1.print("  WELCOME");
    delay(3000); // Wait for 3000 millisecond(s)
    lcd_1.clear();
}
```

```
void loop()
{
    // READ DISTANCE FROM ULTRASONIC SENSORS
    distSensor1 = 0.01723 * readUltrasonicDistance(A3, A2);
    distSensor2 = 0.01723 * readUltrasonicDistance(A1, A0);
    // DETECT INCOMING
    if (distSensor1 < 50) {
        flag1 = 1;
        if (flag2 == 0) {
            person = (person + 1);
            delay(1000); // Wait for 1000 millisecond(s)
        }
    }
}
```

```

// DETECT OUTGOING
if (distSensor2 < 50) {
  flag2 = 1;
  if (flag1 == 0) {
    if (person >= 1) {
      person = (person - 1);
      delay(1000); // Wait for 1000 millisecond(s)
    }
  }
}
// SET FLAGS TO ZERO
if (distSensor1 > 50 && (distSensor2 > 50 && (flag1 == 1
&& flag2 == 1))) {
  flag1 = 0;
  flag2 = 0;
}
// DISPLAY IN LCD SCREEN
lcd_1.setCursor(0, 0);
lcd_1.print("HAVE PERSON ");
if (person < 10) {
  lcd_1.print(person);
  lcd_1.print(" ");
} else {
  if (person < 15) {
    lcd_1.print(person);
    lcd_1.print(" ");
  } else {
    lcd_1.print("MAX");
    person = 15;
    tone(2, 548668578, 1000); // play tone 300 (C25 =
548668578 Hz)
    delay(500); // Wait for 500 millisecond(s)
  }
}

```

```
}  
lcd_1.setCursor(0, 1);  
lcd_1.print("LIGHT IS ");  
if (person >= 1) {  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, HIGH);  
    lcd_1.print("ON ");  
    if (person >= 5) {  
        digitalWrite(4, LOW);  
        digitalWrite(5, LOW);  
        digitalWrite(6, HIGH);  
        if (person >= 10) {  
            digitalWrite(4, LOW);  
            digitalWrite(5, LOW);  
            digitalWrite(6, LOW);  
        }  
    }  
}  
} else {  
    digitalWrite(4, HIGH);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, HIGH);  
    lcd_1.print("OFF");  
}  
}
```