Tank Level Control System

Problem Statement for a Tank Level Control System

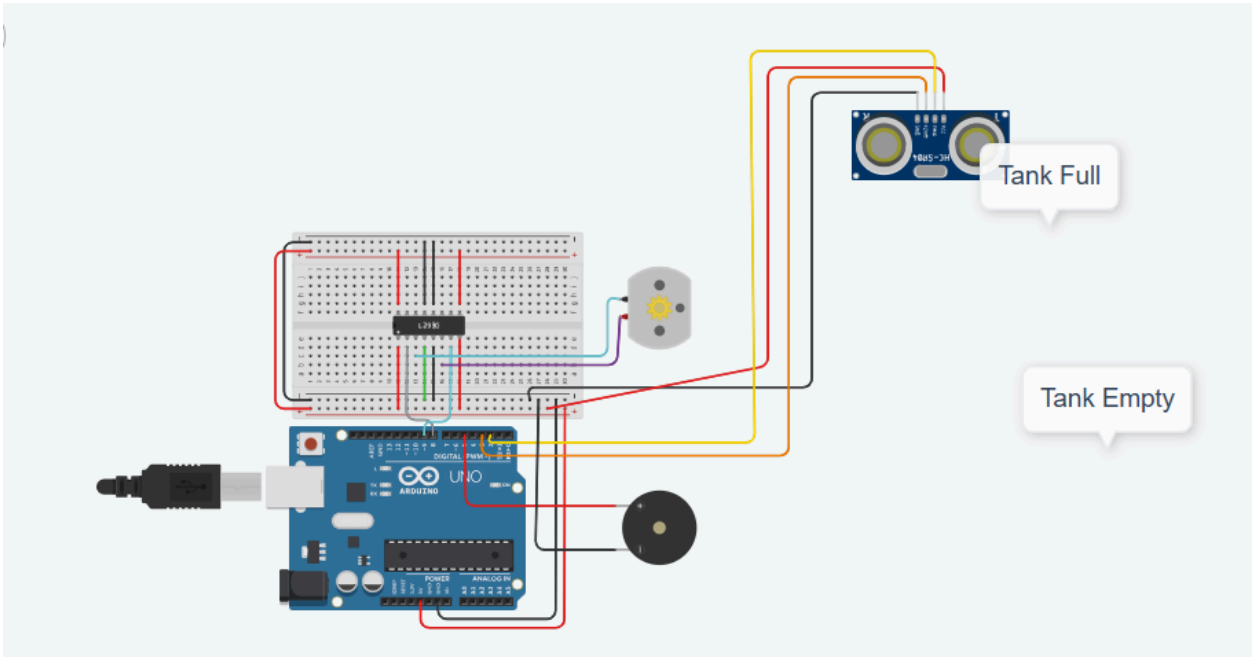
Problem Statement:
In industrial settings, storage tanks are used to hold various liquids essential for production processes. The challenge lies in maintaining the liquid level within these tanks within a predefined range to ensure process stability, prevent hazards such as overflows or dry-running, and optimize resource utilization. Manual monitoring and control of tank levels are prone to human error, leading to inefficiencies, safety risks, and potential downtime.

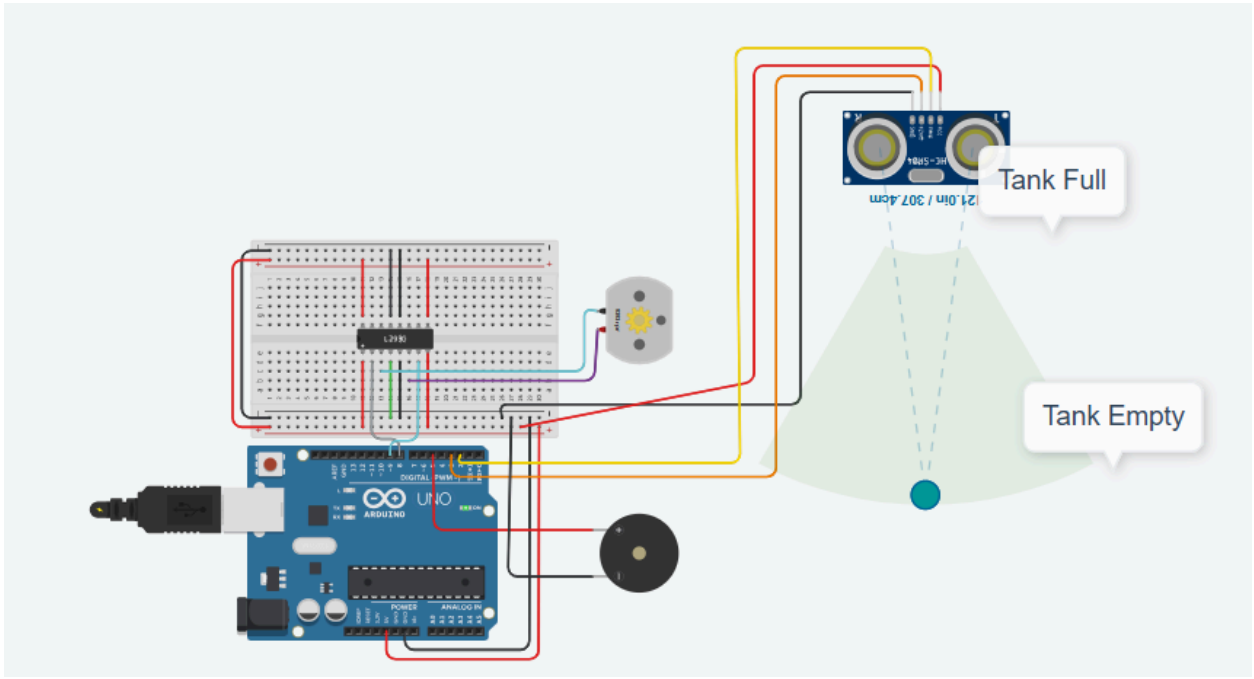
An automated Tank Level Control System is required to continuously monitor the liquid level, adjust the inflow and outflow rates in real-time, and trigger alarms or corrective actions when the levels approach critical thresholds. The system must be reliable, capable of integrating with existing industrial control systems, and adaptable to different types of liquids and tank sizes. It should also minimize operational costs while ensuring safety and compliance with regulatory standards.

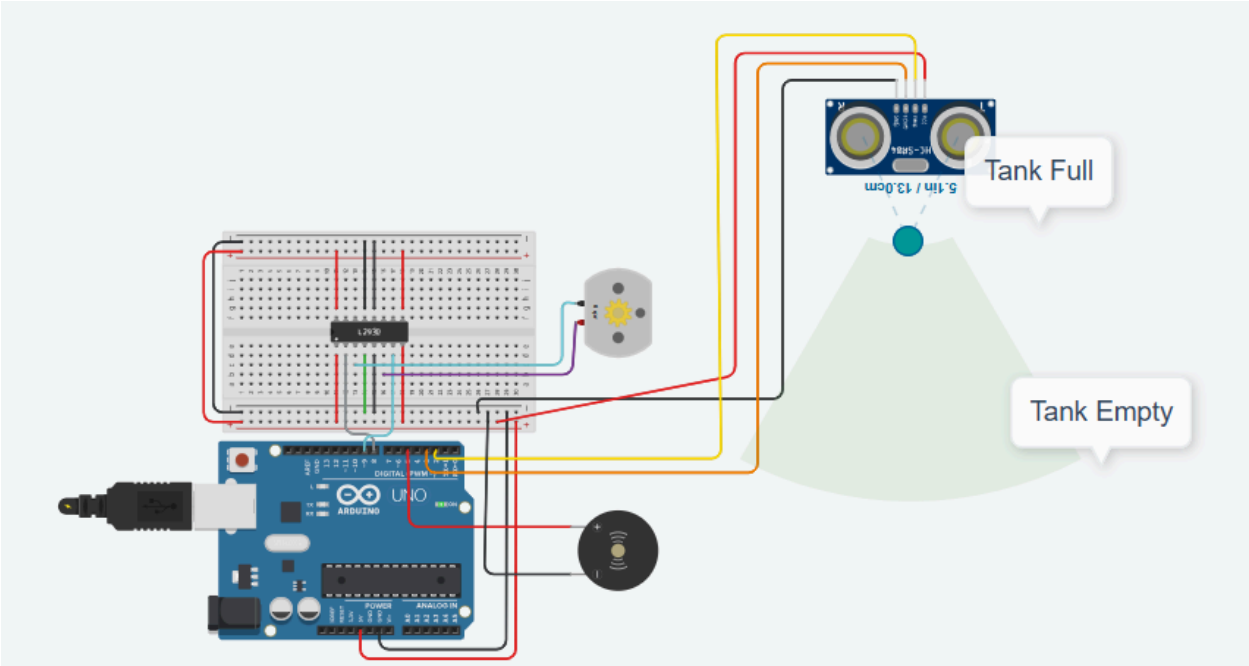
Purpose of a Tank Level Control System

A Tank Level Control System is designed to monitor and regulate the liquid level within a storage tank, ensuring that it remains within a desired range. The primary purposes of such a system include:

1. **Maintaining Desired Liquid Levels:** The system ensures that the liquid level in the tank does not exceed or fall below specific thresholds, preventing overflow or drying out, which could lead to system failures or hazards.
2. **Process Stability:** In industrial processes, maintaining a consistent liquid level is crucial for stable operation. Fluctuations in the liquid level can affect the quality and efficiency of the process, making precise control essential.
3. **Safety:** By preventing overfilling or underfilling, the system reduces the risk of accidents, such as spillage of hazardous materials or damage to equipment.
4. **Automation and Efficiency:** Automating the control of liquid levels in tanks reduces the need for manual monitoring, improving operational efficiency and reducing the potential for human error.
5. **Resource Management:** It helps in managing the resources effectively, ensuring that the necessary amount of liquid is available for processes without wastage.


Tank Full
Tank Empty


Tank Full
Tank Empty



Name	Quantity	Component
U2	1	Arduino Uno R3
DIST1	1	Ultrasonic Distance Sensor (4-pin)
M1	1	DC Motor
PIEZO1	1	Piezo
U3	1	H-bridge Motor Driver

Code:

```
int LevelSensorVal = 0;

int echoPin = 0;

int triggerPin = 0;

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()
{
  Serial.begin(9600);

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop()
{
  echoPin = 3;
  triggerPin = 2;
  LevelSensorVal = 0.01723 * readUltrasonicDistance(2, 3);
  Serial.println("Level Control System");
  Serial.println(LevelSensorVal);
  if (LevelSensorVal <= 40) {
    Serial.println("Tank is FULL");
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    tone(5, 19, 1000); // play tone 3 (D#0 = 19 Hz)
  }
  if (LevelSensorVal >= 300) {
    Serial.println("Tank is EMPTY");
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```

VOTING MACHINE

Problem Statement for a VOTING MACHINE System

Problem Statement:
A problem state of a voting machine can refer to various issues that might prevent it from functioning correctly or securely during an election. Common problems include:

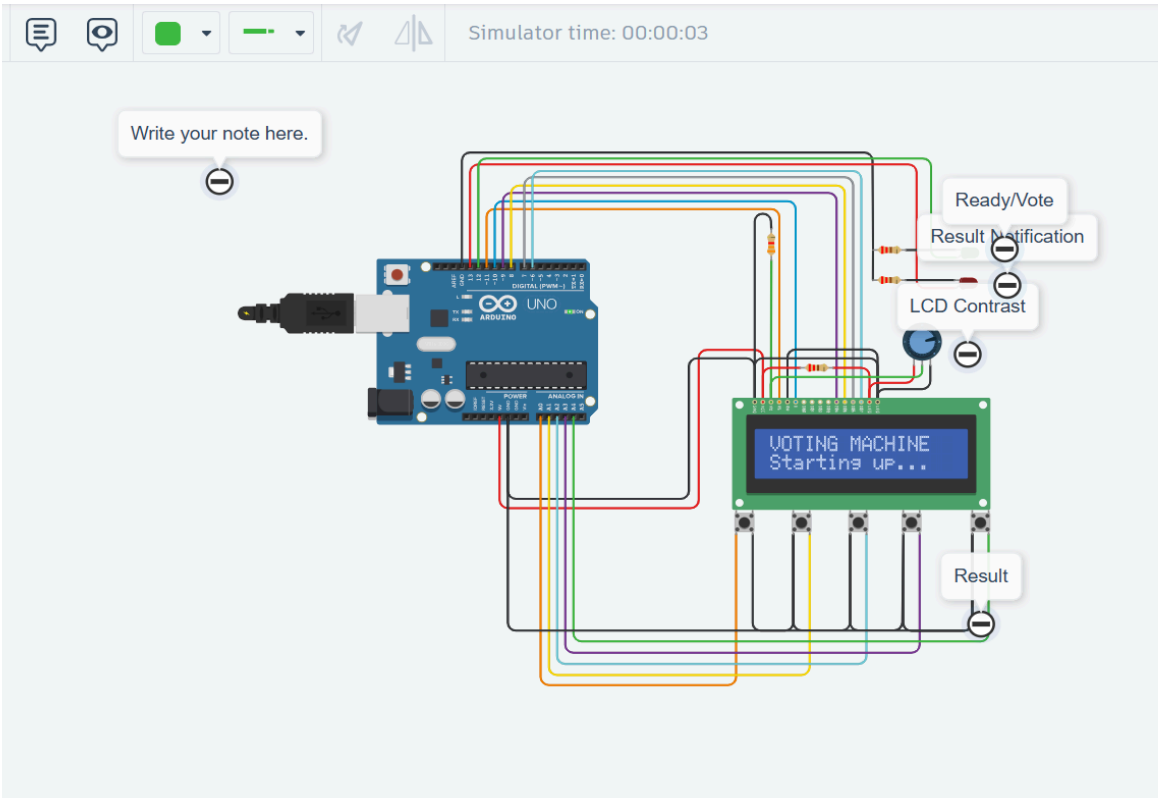
1. **Hardware Malfunctions**:
    - **Touchscreen Issues**: Touchscreens may not respond correctly or register the wrong choice.
    - **Printer Problems**: If a voting machine includes a printer to produce a paper trail, jams, or malfunctions can occur.
    - **Power Failures**: Machines may lose power or have battery issues, especially if backup power systems are not in place.
    - **Memory Card Failures**: Problems with the memory cards used to store votes can result in data loss or corruption.
2. **Software Glitches**:
    - **Freezing or Crashing**: The software on the voting machine may freeze or crash, leading to lost votes or delayed voting.
    - **Incorrect Totals**: Errors in the vote counting software can cause incorrect totals to be displayed or reported.
    - **UI/UX Issues**: Poorly designed interfaces can lead to voter confusion or mistakes.
3. **Security Vulnerabilities**:
    - **Unauthorized Access**: If not properly secured, voting machines could be tampered with or hacked.
    - **Network Vulnerabilities**: Machines connected to a network could be exposed to external threats if the network is not secured.
4. **Calibration Issues**:
    - **Misalignment**: If the screen or input devices are not properly calibrated, voters might inadvertently select the wrong candidates.
5. **Setup and Configuration Errors**:
    - **Incorrect Ballot Layout**: A machine might display an incorrect ballot due to configuration errors.
    - **Wrong Date/Time Settings**: Misconfigured date/time settings can affect the timing of vote processing.
6. **Lack of Accessibility**:
    - **Accessibility Features Not Working**: Machines may fail to provide necessary accessibility options for voters with disabilities.
7. **Operational Issues**:
    - **Improper Training**: Poll workers might not be properly trained to handle the machines, leading to operational problems.

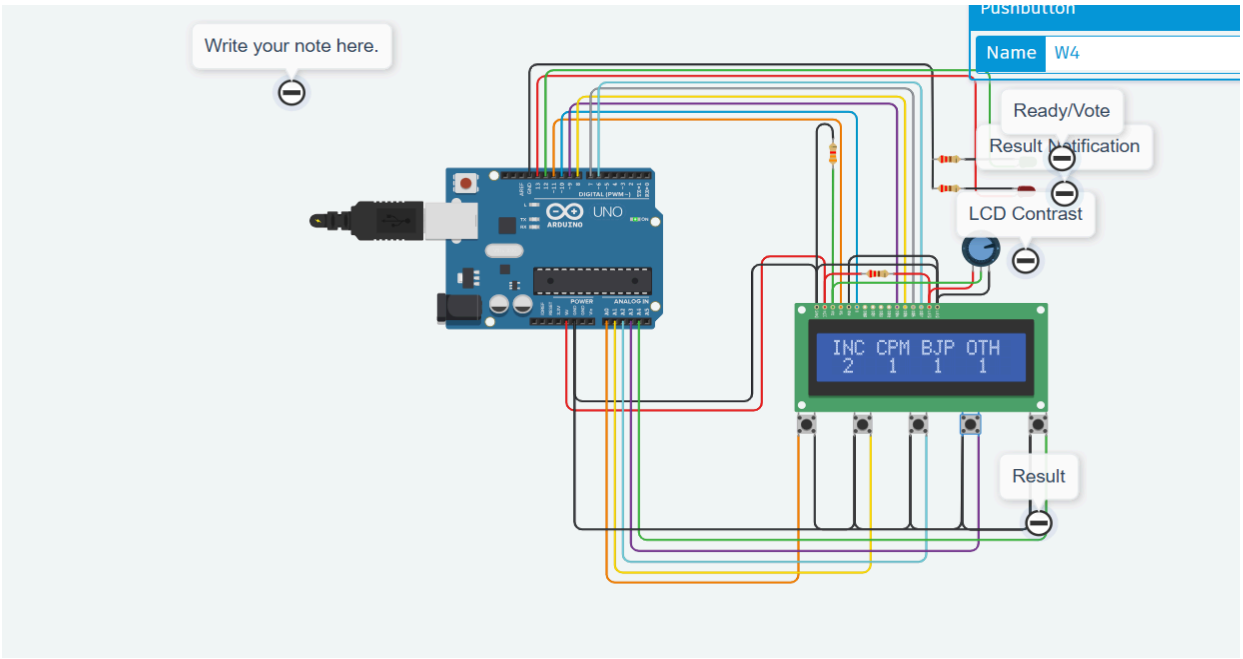
- **Long Boot Times**: Machines that take a long time to start up can delay the voting process, especially during peak times.

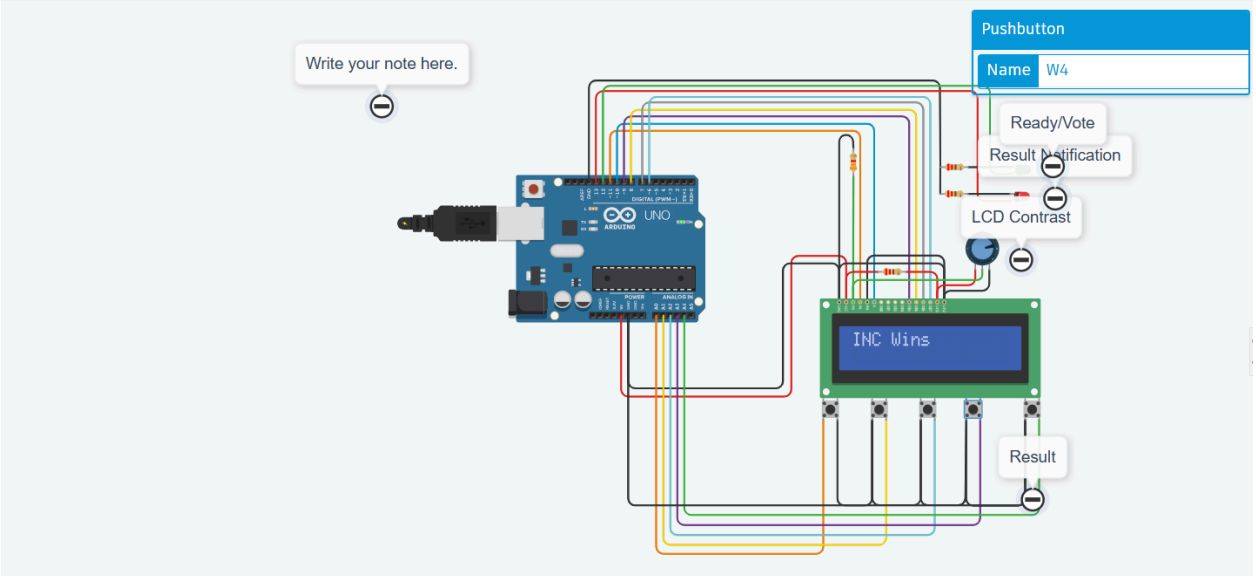
Purpose of a VOTING MACHINE System

The purpose of a voting machine is to facilitate the process of casting, recording, and counting votes in an election. Voting machines aim to improve the efficiency, accuracy, and accessibility of the voting process. Here are the primary purposes of a voting machine:

1. **Accuracy**: Voting machines are designed to accurately capture and count each voter's choice, reducing the chances of human error that can occur with paper ballots.
2. **Efficiency**: Voting machines streamline the voting process by allowing voters to cast their ballots quickly. This can help reduce wait times and manage large volumes of voters more effectively.
3. **Security**: Modern voting machines incorporate various security measures to protect against tampering, unauthorized access, and fraud. They ensure that the votes cast are securely recorded and that the results are trustworthy.
4. **Accessibility**: Voting machines can be equipped with features that make voting more accessible to individuals with disabilities, such as audio assistance, Braille interfaces, or touchscreens.
5. **Auditability**: Many voting machines produce a paper trail or electronic record that can be used to audit and verify the results of an election, ensuring transparency and confidence in the electoral process.
6. **Confidentiality**: Voting machines help maintain voter privacy by allowing individuals to cast their votes without revealing their choices to others.
7. **Speed**: Automated counting systems in voting machines can provide faster results compared to manual counting, allowing election results to be tallied and reported more quickly.
8. **Consistency**: By standardizing the voting process, machines ensure that all votes are counted consistently, regardless of where they are cast.


Simulator time: 00:00:03
Write your note here.
Ready/Vote
Result Notification
LCD Contrast
VOTING MACHINE
Starting up...
Result


Pushbutton
Name W4
Write your note here.
Ready/Vote
Result Notification
LCD Contrast
INC CPM BJP OTH
2 1 1 1
Result



Name	Quantity	Component
UArduino	1	Arduino Uno R3
ULCD	1	LCD 16 x 2
D1	1	Red LED
D2	1	Green LED
R1 R2 R4	3	220 Ω Resistor
R3	1	3.3 kΩ Resistor
SW1 SW2 SW3 SW4 SW5	5	Pushbutton
Rpotlcd contrast	1	10 kΩ Potentiometer

CODE:-

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(11,10,9,8,7,6);
#define sw1 A0
#define sw2 A1
#define sw3 A2
#define sw4 17
#define sw5 18
int vote1=0;
int vote2=0;
int vote3=0;
int vote4=0;
void setup()
{
pinMode(sw1, INPUT);
pinMode(sw2,INPUT);
pinMode(sw3,INPUT);
pinMode(sw4,INPUT);
pinMode(sw5,INPUT);
pinMode(13,OUTPUT);
pinMode(12,OUTPUT);

lcd.begin(16, 2);
lcd.setCursor(0,1);
lcd.print("Please wait...");
lcd.setCursor(0,0);
delay(3000);
lcd.print("VOTING MACHINE");
lcd.setCursor(0,1);
lcd.print("Starting up...");
delay(3000);
lcd.setCursor(0,0);
lcd.print("  Created By  ");
lcd.setCursor(0,1);
lcd.print("sana and salma ");
delay(3000);
digitalWrite(sw1, HIGH);
digitalWrite(sw2, HIGH);
digitalWrite(sw3, HIGH);
digitalWrite(sw4, HIGH);
digitalWrite(sw5, HIGH);
```

```
lcd.clear();
lcd.setCursor(0,0);
lcd.print("INC");
lcd.setCursor(4,0);
lcd.print("CPM");
lcd.setCursor(8,0);
lcd.print("BJP");
lcd.setCursor(12,0);
lcd.print("OTH");
}
void loop()
{
lcd.setCursor(0,0);
lcd.print("INC");
lcd.setCursor(1,1);
lcd.print(vote1);
lcd.setCursor(4,0);
lcd.print("CPM");
lcd.setCursor(5,1);
lcd.print(vote2);
lcd.setCursor(8,0);
lcd.print("BJP");
lcd.setCursor(9,1);
lcd.print(vote3);
lcd.setCursor(12,0);
lcd.print("OTH");
lcd.setCursor(13,1);
lcd.print(vote4);

if(digitalRead(sw1)==0)
{ digitalWrite(12,HIGH);
vote1++;
while(digitalRead(sw1)==0);
digitalWrite(12,LOW);
delay(1000);

}

if(digitalRead(sw2)==0)
{

vote2++;
digitalWrite(12,HIGH);
while(digitalRead(sw2)==0);
```

```
digitalWrite(12,LOW);
delay(1000);

}

if(digitalRead(sw3)==0)
{
digitalWrite(12,HIGH);
vote3++;
while(digitalRead(sw3)==0);
digitalWrite(12,LOW);
delay(1000);
}

if(digitalRead(sw4)==0)
{
digitalWrite(12,HIGH);
vote4++;
while(digitalRead(sw4)==0);
digitalWrite(12,LOW);
delay(1000 );
}

if(digitalRead(sw5)==0)
{
digitalWrite(13,HIGH);
int vote=vote1+vote2+vote3+vote4;
if(vote)
{
if((vote1 > vote2 && vote1 > vote3 && vote1 > vote4))
{
lcd.clear();
lcd.print("INC Wins");
delay(5000);
lcd.clear();
}
else if((vote2 > vote1 && vote2 > vote3 && vote2 > vote4))
{
lcd.clear();
lcd.print("CPM Wins");
delay(5000);
lcd.clear();
}
else if((vote3 > vote1 && vote3 > vote2 && vote3 > vote4))
```

```
{
lcd.clear();
lcd.print("BJP Wins");
delay(5000);
lcd.clear();
}
else if(vote4 > vote1 && vote4 > vote2 && vote4 > vote3)
{
lcd.setCursor(0,0);
lcd.clear();
lcd.print("OTH Wins");
delay(5000);
lcd.clear();
}

else if(vote4 > vote1 && vote4 > vote2 && vote4 > vote3)
{
lcd.setCursor(0,0);
lcd.clear();
lcd.print("OTH Wins");
delay(2000);
lcd.clear();
}

else
{
lcd.clear();
lcd.print("Tie Up Or");
lcd.setCursor(0,1);
lcd.print("No Result");
delay(5000);
lcd.clear();
}

}
else
{
lcd.clear();
lcd.print("Vote first!");
delay(5000);
lcd.clear();
}
vote1=0;vote2=0;vote3=0;vote4=0,vote=0;
lcd.clear();
```

```
digitalWrite(12,HIGH);
digitalWrite(13,LOW);
}
}
```