### ****Voting Machine Algorithm****

#### ****1. Initialization****

* **Initialize GPIO Pins** for voting input buttons:
  + DI1: Vote for Candidate 1
  + DI2: Vote for Candidate 2
  + DI3: Vote for Candidate 3
  + DI4: Vote for Candidate 4
  + DI8: End Voting and Display Results
* **Initialize Serial Communication** displaying instructions.
* **Initialize Vote Counters**:
  + candidate\_1\_votes = 0
  + candidate\_2\_votes = 0
  + candidate\_3\_votes = 0
  + candidate\_4\_votes = 0
  + total\_votes = 0
  + voting\_status = false (indicates if voting is complete)
* **Display Voting Instructions** on Serial Monitor:
  + Show user how to vote for each candidate and how to end the voting process.

#### ****3. Start Voting Process****

**Display Voting Instructions**:

--- Voting Machine Interface ---

Press the corresponding button to vote:

DI1 -> Vote for Candidate 1

DI2 -> Vote for Candidate 2

DI3 -> Vote for Candidate 3

DI4 -> Vote for Candidate 4

DI8 -> End Voting and Show Results

---------------------------------

**Main Voting Loop**:

* + Continuously monitor button presses (DI1 to DI8):
    - If **DI1** is pressed, increment candidate\_1\_votes and total\_votes.
    - If **DI2** is pressed, increment candidate\_2\_votes and total\_votes.
    - If **DI3** is pressed, increment candidate\_3\_votes and total\_votes.
    - If **DI4** is pressed, increment candidate\_4\_votes and total\_votes.
    - If **DI8** is pressed, set voting\_status = true to end the voting process and stop the voting loop.

#### ****4. End Voting and Display Results****

* If voting\_status == true (indicating voting is complete):
* **Display Voting Results** on the Serial Monitor:

--- Voting Results ---

Candidate 1 Votes: <candidate\_1\_votes>

Candidate 2 Votes: <candidate\_2\_votes>

Candidate 3 Votes: <candidate\_3\_votes>

Candidate 4 Votes: <candidate\_4\_votes>

Total Votes: <total\_votes>

#### ****5. Reset Voting ()****

* **Wait for a brief period** (e.g., 5 seconds) to display the results