**Pseudocode for PyPoll Challenges (Rough work)**

To help with this task, please see the pseudocode, logical expression and explanation and a textual flow chart for the calculations only.

1. **The total number of votes cast**

Set a variable TotalVotes to 0. For every row, if value of cell is not empty:

TotatlVotes = TotalVotes + cell value

In python the code will be:

For row in CSV File:

TotalVotes += 1

And print, TotalVotes using concatenation as below:

print(f"Total Votes: {TotalVotes}")

1. **A complete list of candidates who received votes**

To get this, set an empty variable array and create a dictionary known as CandidateVotes, ‘CondidateVotes = {}’ Here the keys are the names of the candidates and values are numbers of votes each candidate received.

Then, set the column 3 as a search row for candidates using ‘candidate name = row[2]’ as third row in the excel sheet contains the list of the candidates’ names.

Now the name is picked up from the list, which means at least one vote was received by this candidate.

If the candidate’s name is in the candidate Votes array list (which checks for the repeated votes for same name), then

CavdidateVote (candidate) = CavdidateVote (candidate) + 1

Otherwise,

CavdidateVote (candidate) = 1

1. **The percentage of votes each candidate won and The total number of votes each candidate won**

First we need to find the votes. We have calculated candidateVotes, so far, where the key was candidate and the value was number of votes received. Now find the percentage:

For votes in each candidateVotes:

P=( Votes/TotalVotes)\*100 (joining first and second question)

Then print the list of the candidates and the votes they received:-

print(f"{candidate}: {percentage:.3f} % ({Votes})")

Candidate is the name of candidate, percent prints value of p, .’3f’ prints 3 decimal points, and % signs and total number of votes for that candidate.

1. **The winner of the election based on popular vote**

To calculate the winner, we find the find the maximum number of candidate votes received by each candidate using:

winner = max(candidateVotes, key=candidateVotes.get)

The key argument is applicable to a function that will be applied to each element, to compare them.

‘candidateVotes.get’ is a method we can use to returns the value ( the number of votes for each candidate.

Using the ‘candidate\_votes.get’ as the key function, max() will compare candidates based on the number of votes they received and find the highest value to find the winner.

**Textual Flowchart for the pseudocode above**

**Start**

**Initialize:** TotalVotes = 0, CandidateVotes = {}

**Loop:** For each row in the CSV file:

* **Increment:** TotalVotes += 1
* **Extract:** candidate = row[2]
* **Check:**
  + **If candidate is in CandidateVotes:** CandidateVotes[candidate] += 1
  + **Else:** CandidateVotes[candidate] = 1

**Loop End**

**Calculate Percentage:** For each candidate in CandidateVotes:

* percentage = (Votes / TotalVotes) \* 100

**Find Winner:** winner = max(CandidateVotes, key=CandidateVotes.get)

**Print Results:**

* TotalVotes
* Each candidate's percentage and votes
* winner

**End**