

```

# The data we need to retrieve.
# 1. The total number of votes cast
# 2. A complete list of candidates who received votes
# 3. The percentage of votes each candidate won
# 4. The total number of votes each candidate won
# 5. The winner of the election based on popular vote.

import csv
import os #os module to access file which path unknown

# Assign a variable for the file to load and the path. #CHAINING
file_to_load = os.path.join("Resources", "election_results.csv")

# Create a filename variable to a direct or indirect path to the file.
file_to_save = os.path.join('analysis' , 'election_analysis.txt')

# Initialize a total vote counter
total_votes = 0

# Candidate Options
candidate_options = []

# Declare the empty dictionary
candidate_votes = {}

# Winning Candidate and Winning Count Tracker
winning_candidate = ""
winning_count = 0
winning_percentage = 0

# Open the election results and read the file.
with open(file_to_load) as election_data :
    file_reader = csv.reader(election_data)

    # Print the header row.
    headers = next(file_reader)

    #Print each row in the CSV file
    for row in file_reader :
        # Add to the total vote count
        total_votes += 1

        # Print the candidate name from each row
        candidate_name = row[2]

        # If the candidate does not match any existing candidate
        # add it to the candidate list.
        if candidate_name not in candidate_options :

```

```

        # Add it to the candidate list
        candidate_options.append(candidate_name)

        # Begin tracking that candidate's vote count
        candidate_votes[candidate_name] = 0

    # Add a vote to that candidate's count
    candidate_votes[candidate_name] += 1

# Save the results to our text file
with open(file_to_save, "w") as txt_file :
    # Print the final vote count to the terminal
    election_results = (
        f"\nElection Results\n"
        f"-----\n"
        f"Total Votes: {total_votes:,}\n"
        f"-----\n")
    print(election_results, end="")

    # Save the final vote count to the text file.
    txt_file.write(election_results)

    # Determine the percentage of votes for each candidate by looping through the counts
    # Iterate through the candidate list
    for candidate_name in candidate_votes :
        # Retrieve vote count of a candidate
        votes = candidate_votes[candidate_name]
        # Calculate the percentage of votes
        vote_percentage = float(votes) / float(total_votes) * 100

        candidate_results = (f"{candidate_name}: {vote_percentage:.1f}% ({votes:,})\n")

        # Print each candidate, their voter count, and percentage to the terminal.
        print(candidate_results)
        # Save the candidate results to our text file.
        txt_file.write(candidate_results)

        # To do: print out each candidate's name, vote count, and percentage of
        # votes to the terminal.
    # print(f"{candidate_name}: {vote_percentage:.1f}% {votes:,}\n")

    # Determine the winning vote count and candidate
    # Determine if the votes are greater than the winning count
    if (votes > winning_count) and (vote_percentage > winning_percentage) :
        winning_count = votes
        winning_candidate = candidate_name
        winning_percentage = vote_percentage

    # Print the winning candidate's results to the terminal
    winning_candidate_summary = (

```

```
f"-----\n"  
f"Winner: {winning_candidate}\n"  
f"Winning Vote Count: {winning_count:,}\n"  
f"Winning Percentage: {winning_percentage:.1f}%\n"  
f"-----\n")
```

```
print(winning_candidate_summary)
```

```
# Save the winning candidate's results to the text file.  
txt_file.write(winning_candidate_summary)
```