# 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 populare 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 wining 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)