"C:/Users/Mary/Data Visualizations BC/Election\_Analysis/election\_results.csv"

"Charles Casper Stockham", "Diana DeGette", "Raymon Anthony Doane"

# 3. Print the total votes.

print(total\_votes)

# Print the candidates list.

print(candidate\_options)

#Print the candidate vote dictionary.

print(candidate\_votes)

# Add our dependencies.

import csv

import os

# Assign a variable to load a file from a path.

file\_to\_load = os.path.join("C:/Users/Mary/Data Visualizations BC/Election\_Analysis/election\_results.csv")

#create a filename variable to a direct or indirect path to the file.

file\_to\_save = os.path.join("C:/", "election\_analysis.txt")

# 1. Initialize a total vote counter.

total\_votes = 0

# candidate options.

candidate\_options = []

# candidate votes.

candidate\_votes = {}

# Open the election results and read the file.

with open(file\_to\_load) as election\_data:

    # To do: read and analyze the data here.

    file\_reader = csv.reader(election\_data)

    # Read the header row.

    headers = next(file\_reader)

    # Print each row in the CSV file.

    for row in file\_reader:

        # 2. 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...

    if candidate\_name not in candidate\_options:

            # Add it to the list of candidates.

            candidate\_options.append(candidate\_name)

            # Begin tracking the candidate's vote count.

            candidate\_votes[candidate\_name] = 0

# Add our dependencies.

import csv

import os

# Assign a variable to load a file from a path.

file\_to\_load = os.path.join("C:/Users/Mary/Data Visualizations BC/Election\_Analysis/election\_results.csv")

#create a filename variable to a direct or indirect path to the file.

file\_to\_save = os.path.join("C:/", "election\_analysis.txt")

# 1. Initialize a total vote counter.

total\_votes = 0

# candidate options.

candidate\_options = []

# candidate votes.

candidate\_votes = {}

# Open the election results and read the file.

with open(file\_to\_load) as election\_data:

    # To do: read and analyze the data here.

    file\_reader = csv.reader(election\_data)

    # Read the header row.

    headers = next(file\_reader)

    # Print each row in the CSV file.

    for row in file\_reader:

        # 2. 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...

    if candidate\_name not in candidate\_options:

            # Add it to the list of candidates.

            candidate\_options.append(candidate\_name)

            # Begin tracking the candidate's vote count.

            candidate\_votes[candidate\_name] = 0

#Print the candidate vote dictionary.

print(candidate\_votes)

# Print the candidate vote dictionary.

print(candidate\_votes)

    # 4. Print the candidate name and percentage of votes.

    print(f'{candidate\_name}: received {vote\_percentage}% of the vote.')

# Add our dependencies.

import csv

import os

# Assign a variable to load a file from a path.

file\_to\_load = os.path.join("C:/Users/Mary/Data Visualizations BC/Election\_Analysis/election\_results.csv")

# Assign a variable to save the file to a path.

file\_to\_save = os.path.join("C:", "election\_analysis.txt")

# Initialize a total vote counter.

total\_votes = 0

# Candidate options and candidate votes

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)

    # Read 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 candidate\_name not in candidate\_options:

            # Add the candidate name 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

# Determine the percentage of votes for each candidate by looping through the counts.

# 1. Iterate through the candidate list.

for candidate\_name in candidate\_votes:

    # 2. Retrieve vote count of a candidate.

    votes = candidate\_votes[candidate\_name]

    # 3. Calculate the percentage of votes.

    vote\_percentage = round(float(votes) / float(total\_votes) \* 100, 1)

# 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

    #  To do: print out each candidate's name, vote count, and percentage of votes to the terminal.

    # Determine winning vote count and candidate

    # Determine if the votes is greater than the winning count.

    if (votes > winning\_count) and (vote\_percentage > winning\_percentage):

         # If true then set winning\_count = votes and winning\_percent =

         # vote\_percentage.

         winning\_count = votes

         winning\_percentage = vote\_percentage

         # And, set the winning\_candidate equal to the candidate's name.

         winning\_candidate = candidate\_name

    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")

#  To do: print out the winning candidate, vote count and percentage to terminal.

print(f"{candidate\_name}: {vote\_percentage:.1f}% ({votes:,})\n")

print(winning\_candidate\_summary)

    # Read 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 candidate\_name not in candidate\_options:

           # Add the candidate name to the candidate list.

            candidate\_options.append(candidate\_name)

           # 2. 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

# Add our dependencies.

import csv

import os

# Assign a variable to load a file from a path.

file\_to\_load = os.path.join("C:/Users/Mary/Data Visualizations BC/Election\_Analysis/election\_results.csv")

# Assign a variable to save the file to a path.

file\_to\_save = os.path.join("C:", "election\_analysis.txt")

# Initialize a total vote counter.

total\_votes = 0

# Candidate options and candidate votes

candidate\_options = []

# 1. 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)

    # Read 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 candidate\_name not in candidate\_options:

            # Add the candidate name to the candidate list.

            candidate\_options.append(candidate\_name)

            # 2. 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

# Determine the percentage of votes for each candidate by looping through the counts.

# 1. 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 = round(float(votes) / float(total\_votes) \* 100, 1)

    # Print the candidate name and percentage of votes.

    # Determine winning vote count and candidate

    # Determine if the votes are greater than the winning count.

    if (votes > winning\_count) and (vote\_percentage > winning\_percentage):

         # If true then set winning\_count = votes and winning\_percent = vote\_percentage.

        winning\_count = votes

        winning\_percentage = vote\_percentage

        # Set the winning\_candidate equal to the candidate's name.

        winning\_candidate = candidate\_name

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")

# 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")

 # Print the winning candidate's summary.

print(winning\_candidate\_summary)

 print(election\_results, end="")

    # Save the final vote count to the text file.

    txt\_file.write(election\_results)