# -\*- coding: UTF-8 -\*-

"""PyPoll Homework Challenge Solution."""

# Add our dependencies.

import csv

import os

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

file\_to\_load = os.path.join("..", "Resources", "election\_results.csv")

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

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

# Initialize a total vote counter.

total\_votes = 0

# Candidate Options and candidate votes.

candidate\_options = []

candidate\_votes = {}

# 1: Create a county list and county votes dictionary.

# Track the winning candidate, vote count and percentage

winning\_candidate = ""

winning\_count = 0

winning\_percentage = 0

# 2: Track the largest county and county voter turnout.

# Read the csv and convert it into a list of dictionaries

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

reader = csv.reader(election\_data)

# Read the header

header = next(reader)

# For each row in the CSV file.

for row in reader:

# Add to the total vote count

total\_votes = total\_votes + 1

# Get the candidate name from each row.

candidate\_name = row[2]

# 3: Extract the county name from each row.

# If the candidate does not match any existing candidate add it to

# the candidate list

if candidate\_name not in candidate\_options:

# Add the candidate name to the candidate list.

candidate\_options.append(candidate\_name)

# And begin tracking that candidate's voter count.

candidate\_votes[candidate\_name] = 0

# Add a vote to that candidate's count

candidate\_votes[candidate\_name] += 1

# 4a: Write an if statement that checks that the

# county does not match any existing county in the county list.

# 4b: Add the existing county to the list of counties.

# 4c: Begin tracking the county's vote count.

# 5: Add a vote to that county's vote count.

# Save the results to our text file.

with open(file\_to\_save, "w") as txt\_file:

# Print the final vote count (to terminal)

election\_results = (

f"\nElection Results\n"

f"-------------------------\n"

f"Total Votes: {total\_votes:,}\n"

f"-------------------------\n\n"

f"County Votes:\n")

print(election\_results, end="")

txt\_file.write(election\_results)

# 6a: Write a for loop to get the county from the county dictionary.

# 6b: Retrieve the county vote count.

# 6c: Calculate the percentage of votes for the county.

# 6d: Print the county results to the terminal.

# 6e: Save the county votes to a text file.

# 6f: Write an if statement to determine the winning county and get its vote count.

# 7: Print the county with the largest turnout to the terminal.

# 8: Save the county with the largest turnout to a text file.

# Save the final candidate vote count to the text file.

for candidate\_name in candidate\_votes:

# Retrieve vote count and percentage

votes = candidate\_votes.get(candidate\_name)

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

candidate\_results = (

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

# Print each candidate's voter count and percentage to the

# terminal.

print(candidate\_results)

# Save the candidate results to our text file.

txt\_file.write(candidate\_results)

# Determine winning vote count, winning percentage, and candidate.

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

winning\_count = votes

winning\_candidate = candidate\_name

winning\_percentage = vote\_percentage

# Print the winning candidate (to 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 name to the text file

txt\_file.write(winning\_candidate\_summary)