Test 1: IRV Audit File Checking
Team Member Responsible: Kenji Her
Inputs: 1. ArrayList of candidates 2. ArrayList of ballots 3. Audit file for results 4. Number of seats for the election
Tests: Test if the audit object is created and populated from the algorithm
Outputs: The system will print out the audit object's information if it contains any
Passed or Failed: Passed
Date: 4/22/22

Test 2: ReadCSV Invalid Ballot Checking
Team Member Responsible: Kenji Her
Inputs: 1. The IRV testing CSV file
Tests: Test if the invalid ballots get filtered correctly, and that a file containing the invalid ballots is written
Outputs: The system will print out the valid ballots and invalid ballots, and it will create a file containing the invalid ballots
Passed or Failed: Passed
Date: 4/28/22

Test 3: ReadCSV PO Handling
Team Member Responsible: Daniel
Inputs: None, The path to the PO is declared in the test function
Tests: Test if the PO file is read correctly and the order of ballots is correct. It tests the new else if branch that was created in ReadCSV
Outputs: The test will print out the order of the ballots and will need to be manually compared with expected output.
Passed or Failed: Passed
Date: 4/29/22

Test 4:IRV setup
Team Member Responsible: Rick
Inputs: The ballots data structure from the readCSV method.
Tests: Test to check if the ballots and the order are set up with the correct amount of candidates and votes.
Outputs: It is expected that each candidate has a correct number of ballots with different order of choice.
Passed or Failed: Passed
Date: 5/1/22

Test 5: IRV Result
Team Member Responsible: Rick
Inputs: Path to the ballot file
Tests: The test will make sure that the output of running IRV is correct. The test works by running the main driver and comparing the output to what is expected
Outputs: The results of all of the voting rounds are displayed
Passed or Failed: Passed
Date: 5/1/22

Test 6: IRV 100,000 ballot
Team Member Responsible: Rick
Inputs: Path to the ballot file with 100,000 ballots
Tests: The test will make sure the IRV algorithm can run 100,000 ballots in under 8 minutes
Outputs: The results of all of the voting rounds are displayed
Passed or Failed: Passed
Date: 5/1/22

Test 7: IRV Tie-breaker
Team Member Responsible: Rick
Inputs: Path to the ballot file with a tie
Tests: The test is checking if the algorithm is working correctly for tiebreakers.
Outputs: Each time one out of the last two candidates will be a winner and one will be a loser, but not both.
Passed or Failed: Passed
Date: 5/1/22

Test 8: PO Algorithm Constructor
Team Member Responsible: Rick
Inputs: Path to the ballot file for the PO election
Tests: Test to check if the constructor for the PO class is working and the ballots and the order are set up with the correct amount of candidates and votes.
Outputs: Each candidate has the correct number of ballots.
Passed or Failed: Passed
Date: 5/1/22

Test 9: PO Algorithm Result
Team Member Responsible: Rick
Inputs: Path to the ballot file for the PO election
Tests: The test will make sure that the output of the running PO algorithm is correct. The test works by running the main driver and comparing the output to what is expected
Outputs: The winner is correctly chosen
Passed or Failed: Passed
Date: 5/1/22