Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/26/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_ballot_isEmpty

Test Description:
This test will check if a ballot is empty when it is first created. It will also test if the push() function adds new votes to the ballot.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: isEmpty() and push() from Ballot class

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV election is being ran

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create an empty ballot object	Ballot ballot			
2	Check if ballot is empty	ballot.isEmpty	true	true	
3	Push a candidates ID onto the ballot	ballot.push(3)	false	false	

Post condition(s) for Test:

The ballot has added the appropriate candidates to the ballot and the ballot is no longer empty

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/26/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_ballot_pop

Test Description:
This test will check if the first vote in the ballot is returned.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Function being used: ispop() from Ballot class

Automated: yes__X__ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV election is being ran

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create an empty ballot object	Ballot ballot			
2	Call pop() on the empty ballot	ballot.pop	Segmentation Fault	Segmentation Fault	Calling pop on an empty ballot should create a segmentation fault
3	Push a candidates ID onto the ballot	ballot.push(1)			
4	Call pop() on ballot	ballot.pop	1	1	

Post condition(s) for Test:

The ballot has returned the first vote and it is removed from the ballot

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_candidate_name

Test Description:
This test will check if the correct name is being set for a candidate and check if the candidate's name is being returned.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: setName() and getName() from Candidate class

Automated: yes X no

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a candidate object	Candidate c1			
2	Set the candidate's name	c1.setName("John")			
3	Return the the candidates name	c1.getName()	John	John	
4	Create a candidate without setting a name for the candidate	Candidate c2 c2.getName()	None	None	

Post condition(s) for Test:

The Candidate's name is set up with proper values

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_candidate_addVote

Test Description:
This test will check if votes are being added to a candidate and checking the total votes for a candidate.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: getNumVotes() and addVote() from Candidate class

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a candidate object	Candidate c1			
2	Check if candidate has no votes	c1.getNumVotes()	0	0	Candidate starts with no votes when first initialized
3	Add a vote to the candidate	c1.addVote()			
4	Return the number of votes for the candidate	c1.getNumVotes()	1	1	

Post condition(s) for Test:

The getNumVotes() variable is set up with the proper values

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_candidate_addToParty

Test Description:
This test will check to see if a candidate has been added to a party and the total number of candidates is correct.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: getNumCandidates() , setName() and addVote() from Candidate class

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran. The functions setName() , addVote() and addCandidate() are all working properly.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create an empty party object	Party p			
2	Check if the party has no candidates	p.getNumCandidates()	0	0	
3	Create a new candidate and add it to the party	Candidate* c = new Candidate() c → setName(“Ben”) c → addVote(); p.addCandidate(c)			
4	Check if the number of candidates have been updated	p.getNumCandidates()	1	1	

Post condition(s) for Test:

The candidate has been added to their party and getNumCandidates is set with proper values

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_party_name

Test Description:
This test will check if the correct name is being set for the party and check if the party's name is being returned.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: setName() and getName() from Party class

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a party object	Party p1			
2	Set a party's name	p1.setName("R")			
3	Return the party's name	p1.getName()	R	R	
4	Create a party without setting a name for the party	Party p2 p2.getName	None	None	

Post condition(s) for Test:

The Party's name is set up with proper values

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_party_totalVotes

Test Description:
This test will check the total votes for a party based on votes being added to candidates in the party.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: getTotalVotes() , addCandidate() , addVote(), calculateTotalVotes(), getNumCandidates(), getNumVotes() from Party class

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a party object and candidate objects	Party p Candidate* c1 and Candidate* c2			
2	Return total votes for the party	p.getTotalVotes	0	0	
3	Add candidates to the party and add votes to the candidates	p.addCandidate(c1) and p.addCandidate(c2) c1→ addVote() c2→addVote() , c2→addVote()			
4	Calculate total votes for the party	p.calculateTotalVotes()			
5	Get total votes for the party	p.getTotalVotes()	3	3	

Post condition(s) for Test:

The getTotalVotes() variable is set up with the proper values

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_coinFlip

Test Description:
This test will check if the coin flip is random and unbiased.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: flip() from Coin class

Automated: yes__X_ no ___

Results: Pass X Fail

CreatCr3e

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Initialize a coin and counting variables	Coin c NUM_FLIPS = 300 headCount and tailCount = 0			
2	Calculate the margin of 20% more or less than expected	maxMargin = (NUM_FLIPS / 2) * 1.20 minMargin = (NUM_FLIPS / 2) * 0.80			
3	Initialize a boolean variable for coins fairness	fair = true			
4	record results of 300 flips				
5	check for unusual bias		true	true	

Post condition(s) for Test:

The coin flip is unbiased and random

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_ballotQueue_isEmpty

Test Description:
This test will check if the ballot queue is empty when it is first created. It will also test if ballots are correctly being added to the ballotQueue.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: isEmpty() and push() from BallotQueue class

Automated: yes__X__ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a BallotQueue object	BallotQueue ballotQueue			
2	Create a ballot	Ballot* b1			
3	Check if ballot queue is empty	ballotQueue.isEmpty	true	true	
4	Add the ballot to the ballot queue	ballotQueue.push(b1)			
5	Check if the ballot was added to the ballot queue	ballotQueue.isEmpty()	false	false	

Post condition(s) for Test:

The ballot was added to the ballot queue properly.

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_ballotQueue_count

Test Description:
This test will check if the number of ballots in the ballot queue is correct. Also, it tests if push properly adds ballots to the ballot queue.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: push() and getCount() from BallotQueue class

Automated: yes _X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot queue object	BallotQueue ballotQueue			
2	Create ballots	Ballot* b1, Ballot* b2, Ballot* b3			
3	Push ballots onto the ballot queue	ballotQueue.push(b1) , ballotQueue(b2) , ballotQueue(b3)			
4	Count how many ballots are on the ballot queue	ballotQueue.getCount()	3	3	

Post condition(s) for Test:

The ballot queue has the correct number of ballots

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_ballotQueue_pop

Test Description:
This test will check if ballots are being properly added to the ballot queue. This test will also test if pop() returns the correct ballot.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: push() , pop() and getCount() from BallotQueue class

Automated: yes__X_ no ___

Results: Pass _X_____ Fail_____

Preconditions for Test:
A csv file for the IRV election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot queue object	BallotQueue ballotQueue			
2	Count how many ballots are in the empty ballot queue	ballotQueue.getCount()	0	0	
3	Trying to pop() an empty ballot queue	ballotQueue.pop()	Segmentation Fault	Segmentation Fault	
4	Create ballots	Ballot* b1 , Ballot* b2 , Ballot* b3			
5	Push ballots onto ballot queue	ballotQueue.push(b1) , etc.			
6	Check for the number of ballots in ballot queue	ballotQueue.getCount()	3	3	
7	Return the last element in the queue	ballotQueue.pop()	b3	b3	

Post condition(s) for Test:

The ballot queue has the correct number of elements and the pop() function returns the last element in the queue properly

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_

Test Description:
This test will check if votes are being added to a candidate and checking the total votes for a candidate.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: getNumVotes() and addVote() from Candidate class

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2					
3					
4					

Post condition(s) for Test:

The getNumVotes() variable is set up with the proper values

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit _X_ System __

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: UT_ballotQueue_shuffle

Test Description:
This test will check where the shuffle changes the order of the queue.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File for Test: all_test.cpp
Functions being used: push() , pop() and shuffle() from BallotQueue class

Automated: yes__X_ no ___

Results: Pass _X_____ Fail_____

Preconditions for Test:
A csv file for the IRV election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot queue object	BallotQueue queue			
2	Create ballots	Ballot* b1 , etc.			
3	Push ballots onto the queue	queue.push(b1) , etc.			
4	Shuffle ballots	queue.shuffle()			
5	Pop all ballots	queue.pop , 3 times			
6	Check if the order is the same		true	true	

Post condition(s) for Test:

The shuffle() function works properly

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_01(opl1.csv)

Test Description:
Average election spread. 1 tie between candidates in the same party

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl1.csv

Automated: yes X no

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl1.csv				
2	Enter election data	opl1.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Foster, Pike, and a coin flip between Jones and Borg. The rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_02(opl2.csv)

Test Description:
All candidates have same number of votes. 1 tie between candidates in the same party, 1 tie between two different parties

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl2.csv

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl2.csv				
2	Enter election data	opl2.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Coin flip for one seat between Foster and Pike. And Coin flip for two seats between Borg, Jones and smith. The rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_03(opl3.csv)

Test Description:
One candidates gets all votes, rest of the seats are distributed through coin flip, 1 additional tie in R party.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl3.csv

Automated: yes X no

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl3.csv				
2	Enter election data	opl3.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Coin flip between Foster and pike for one seat. Coin flip between the loser of first coin flip, Borg ,Jones and Smith for two seats

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_04(opl4.csv)

Test Description:
More seats than there are candidates. Everyone gets a seat.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl4.csv

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl4.csv				
2	Enter election data	opl4.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Pike, Borg, and Smith,the rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_05(opl5.csv)

Test Description:
1 ballot, 1 seat, multiple candidates. candidate who got the vote wins.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl5.csv

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl5.csv				
2	Enter election data	opl5.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winner is Pike, the rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_06(opl6.csv)

Test Description:
5 parties, each with 1 candidate, each win one votes. All winners decided not by numbers of votes, but coin flip.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl6.csv

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl6.csv				
2	Enter election data	opl6.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Coin flip for one seat between Foster, Pike, Borg, Jones and Smith. The rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_07(opl7.csv)

Test Description:
3seats, 3 parties with 1 candidate and one vote each, everyone gets a seat.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl7.csv

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl7.csv				
2	Enter election data	opl7.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Pike, Jones and Smith. The rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_08(opl8.csv)

Test Description:
No ballots, with multiple parties, candidates, and seats. All winners decided not by numer of votes, but coin flip.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl8.csv

Automated: yes__X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl8.csv				
2	Enter election data	opl8.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Coin flip for one seat between Foster, Pike, Borg, Jones and Smith. The rest are losers.

Post condition(s) for Test:
Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_09(opl9.csv)

Test Description:
10000 ballots, checking the runtime of opl

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: opl9.csv

Automated: yes X no

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/opl9.csv				
2	Enter election data	opl9.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Pike is the winner. The rest are losers.

Post condition(s) for Test:

Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_01(irv1.csv)

Test Description:
standard election spread.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv1.csv

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv1.csv				
2	Enter election data	irv1.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Rosen and Chou Coin Flip between Royce and Kleinberg

Post condition(s) for Test:
Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_02(irv2.csv)

Test Description:

All candidates except 1 have 0 votes, but there are two seats available. Testing candidate matching up.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv2.csv

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv2.csv				
2	Enter election data	irv2.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Rosen Coin Flip between Kleinberg and Royce

Post condition(s) for Test:
Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_03(irv3.csv)
Test Description:

Example of normal irv. Regardless of shuffle, Rosen should win every time despite having less votes initially. This example works even if you invert the ballots.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv3.csv

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv3.csv				
2	Enter election data	irv3.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winner is Rosen

Post condition(s) for Test:
Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_04(irv4.csv)

Test Description:

Four candidates, four seats, one vote. testing a 3-way tie.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv4.csv

Automated: yes _X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv4.csv				
2	Enter election data	irv4.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Rosen, Kleinberg, Chou and Royce No coin flip

Post condition(s) for Test:
Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_05(irv5.csv)

Test Description:

One candidate, one seat, zero votes, testing if empty ballot queue breaks the system.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv5.csv

Automated: yes _X_ no ___

Results: Pass X Fail

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv5.csv				
2	Enter election data	irv5.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Rosen is winner.

Post condition(s) for Test: Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_06(irv6.csv)
Test Description:

Four candidates, two seats, 24votes that represent all possible permutations. This is capable of testing the actual randomness of our algorithm and shuffle, as the results should be a toss-up

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv6.csv

Automated: yes__X_ no ___

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv6.csv				
2	Enter election data	irv6.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners vary depending on the coin flip

Post condition(s) for Test: Correct results, display, audit file, and media file

Project Name: Project 1: Voting System **Team#** 13

Test Stage: Unit __ System _X_

Test Date: 3/27/22

Name(s) of Testers:
Kaley Schiffler, Amy Nguyen, William Henning, Hoin Jang

Test Case ID#: System Testing_07(irv7.csv)
Test Description:

Four candidates, two seats, 30000 votes. This tests the shuffle and randomness, as rosen gets all the droop # of votes distributed to them first before other votes are distributed, and all permutations are represented equally. Also tests large numbers of votes.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

File directory : Repo-Team13/Project1/testing
File name: irv7.csv

Automated: yes X no

Results: Pass _X____ Fail_____

Preconditions for Test:
A csv file for the IRV or OPL election is being ran.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	run ./build/final_program ../testing/irv7.csv				
2	Enter election data	irv7.csv			
3	Confirm inputs	true			
4	Check if printed display results are correct and that the audit and media files are generated		true	true	Winners are Rosen Coin Flip between Royce and Kleinburg

Post condition(s) for Test: Correct results, display, audit file, and media file