

Project Name: Project 1: Voting System**Team #21****Test Stage:** Unit**Test Date:** March 26, 2023**Test Case ID#:** IR_Ballot_Test_1.0**Name(s) of Testers:** Hady**Test Description:** Testing IR_Ballot functionality**Indicate where are you storing the tests (what file) and the name of the method/functions being used.****Automated:** yes

file: IR_BallotTests.java, method: testIR_Ballot()

Results: Pass**Preconditions for Test:**

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests setBallotNum	.setBallotNum(ball-otNum:1)	1	1	
2	Tests setRank	.setRank(rank:1)	1	1	
3	Tests setForm	.setForm(form:"abc"	"abc"	"abc	
4	Test setCandidateRanking	.setCandidateRank(new int[] {1, 2, 3}	new int[] {1, 2, 3}	new int[] {1, 2, 3}	

Post condition(s) for Test: None

Project Name: Project 1: Voting System**Team# 21****Test Stage:** Unit**Test Date:** March 26, 2023**Test Case ID#:** IR_Candidates_Test_1.0**Name(s) of Testers:** Hady**Test Description:** Testing IR Candidate functionality with a default constructor**Indicate where are you storing the tests (what file) and the name of the method/functions being used.**

file: IR_CandidateTests.java, method: TestIR_Candidate

Automated: yes**Results:** Pass**Preconditions for Test:** Create a new default Candidate object with no set parameters. dflt = new Candidate()

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getBallotCount()	dflt.getBallotCount()	0	0	test run with a new default Candidate constructor like the one in the preconditions
2	Tests getBallots()	dflt.getBallots()	null	null	
3	Tests getName()	dflt.getName()	""	""	
4	Tests getParty()	dflt.getParty()	""	""	

Post condition(s) for Test: None

Project Name: Project 1: Voting System**Team# 21****Test Stage:** Unit**Test Date:** March 26, 2023**Test Case ID#:** IR_Candidates_Test_1.1**Name(s) of Testers:** Hady**Test Description:** Testing the IR_Candidate with a constructor with set parameters**Indicate where are you storing the tests (what file) and the name of the method/functions being used.**

file: IR_CandidateTests.java, method: TestIR_Candidate

Automated: yes**Results:** Pass**Preconditions for Test:** Create a new Candidate, deflt = new Candidate(ballotCount: 3, ballots, name:"Name", party: "Party")

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getBallotCount()	dflt.getBallotCount()	3	3	All tests run with constructor like the one mentioned on preconditions
2	Tests getBallots()	dflt.getBallots()	ballots	ballots	
3	Tests getName()	dflt.getName()	"Name"	"Name"	
4	Tests getParty()	dflt.getParty()	"Party"	"Party"	

Post condition(s) for Test: None

Project Name: Project 1: Voting System**Team# 21****Test Stage:** Unit**Test Date:** March 26, 2023**Test Case ID#:** IR_Candidates_Test_1.2**Name(s) of Testers:** Hady**Test Description:** Testing the setters for the IR_Candidates class.**Indicate where are you storing the tests (what file) and the name of the method/functions being used.**

file: IR_CandidateTests.java, method: TestIR_Candidate

Automated: yes**Results:** Pass**Preconditions for Test:** make an array of ranking[0] with a value of 2 and ballot = new IR_Ballot(ballotNum: 5, rank: 6, form: "abc", rankings)

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests setBallotCount	.setBallotCount(ballotCount: 2)	2	2	
2	Tests setBallots	.setBallots(ballots)	ballots	ballots	
3	Tests setName	.setName("Tyler")	"Tyler"	"Tyler"	
4	Tests setParty	.setParty("R")	"R"	"R"	

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: IR_Election_Integration_Test_1.0

Name(s) of Testers: Andrew Guerra

Test Description: Testing IR audit output files to verify correct format and data

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

file: IR_ElectionIntegrationTests.java, method: testSmallIR_Election

Automated: yes

Results: Pass

Preconditions for Test: None

Step	Test Step Description	Test Data	Expected Result	Actual Result	Notes
------	-----------------------	-----------	-----------------	---------------	-------

#					
1					
2	Ensure small-IR-audit.txt is in testing/IR_ElectionIntegration	new File("../testing/IR_ElectionIntegration/small-IR-audit.txt")	File containing small IR election	File containing small IR election	
3	Run IR election with small-IR.csv and 03-26-2023	Main.main({"small-IR.csv", "03-26-2023"})	No results tested	No results tested	
4	Ensure all lines of small-IR-audit.txt and IR_03-26-2023.txt	small-IR.txt and IR_03-26-2023	All lines are equivalent	All lines are equivalent	Ordering of ballots could be different in audit files due to shuffle but this does not count as non-equivalence

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: IR_Election_Integration_Test_1.1

Name(s) of Testers: Andrew Guerra

Test Description: Testing of IR election with 100,000 ballots for running duration less than 4 minutes

<p>Automated: yes</p>		<p>Indicate where are you storing the tests (what file) and the name of the method/functions being used. file: IR_ElectionIntegrationTests.java, method: testLargeIR_Election</p>
<p>Results: Pass</p>		

<p>Preconditions for Test: None</p>
--

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run IR election with large-IR.csv and 03-26-2023	Main.main({"large-I R.csv", "03-26-2023"})	No result tested	No result tested	
3	Ensure run time is less than 4 minutes	Run time of main	Less than 4 minutes	Less than 4 minutes	
4					

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: CPL_Election_Integration_Test_1.0

Name(s) of Testers: Andrew Guerra

Test Description: Testing CPL audit output files to verify correct format and data

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

file: CPL_ElectionIntegrationTests.java, method: testSmallCPL_Election

Automated: yes

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Ensure small-CPLR-audit.txt is in testing/CPL_ElectionIntegration	new File("../testing/CPL_ElectionIntegration/small-CPL-audit.txt")	File containing small CPL election	File containing small CPL election	
3	Run CPL election with small-CPL.csv and 03-26-2023	Main.main({"small-CPL.csv", "03-26-2023"})	No results tested	No results tested	
4	Ensure all lines of small-CPL-audit.txt and CPL_03-26-2023.txt	small-CPL.txt and CPL_03-26-2023	All lines are equivalent	All lines are equivalent	Ordering of ballots could be different in audit files due to shuffle but this does not count as non-equivalence

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: CPL_Election_Integration_Test_1.1

Name(s) of Testers: Andrew Guerra

Test Description: Testing of CPL election with 100,000 ballots for running duration less than 4 minutes

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

file: CPL_ElectionIntegrationTests.java, method: testLargeCPL_Election

Automated: yes

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					

2	Run IR election with large-IR.csv and 03-26-2023	Main.main({"large-I R.csv", "03-26-2023"})	No result tested	No result tested	
3	Ensure run time is less than 4 minutes	Run time of main	Less than 4 minutes	Less than 4 minutes	
4					

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_AuditTests_1.0

Name(s) of Testers: Hady

Test Description: Testing if the IR_Audit functionality works properly

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

Automated: no

file: IR_AuditTests.java, method: test_IR_Audit_Default

Results: Pass

Preconditions for Test: create a new default IR Audit object, IR_Audit audit = new IR_Audit()

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getDate	audit.getDate()	""	""	
2	Tests getFileName	audit.getFileName()	""	""	
3	Tests getAuditFile	audit.getAuditFile()	null	null	
4	Tests getAudit Writer	audit.getAuditWriter()	null	null	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_AuditTests_1.1

Name(s) of Testers: Hady

Test Description: Testing if the IR_Audit date and getters and setters work properly

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

file: IR_AuditTests.java,

method: test_IR_Audit_Date_getters_and_setters

Automated: no

Results: Pass

Preconditions for Test: create a new audit file, File auditFile = new File("IR_11-23-2002.txt")

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getDate	audit.getDate()	"11-23-2002"	"11-23-2002"	
2	Tests getFileName	audit.getFileName()	"IR_11-23-2002"	"IR_11-23-2002"	
3	Tests getAuditFile	audit.getAuditFileName()	auditFile	auditFile	
4	Tests getAudit Writer	audit.getAuditWriter()	!null	!null	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_AuditTests_1.2

Name(s) of Testers: Hady

Test Description: Testing if the IR_Audit write candidate ballot works properly

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

file: IR_AuditTests.java,

method: test_IR_Audit_writeCandidateBallot

Automated: no

Results: Pass

Preconditions for Test:

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests writeCandidatesBallots	audit. writeCandidatesBallots(candidates, numCandidate: 3)		Actual results match expected results	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_AuditTests_1.3

Name(s) of Testers: Hady

Test Description: Testing if the IR_Audit Ties

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

file: IR_AuditTests.java,

method: test_IR_Audit_LoserTies, test_IR_Audit_WinnerTies

Automated: no

Results: Pass

Preconditions for Test:

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests writeTiedLoserCandidate	audit.writeTiedLoserCandidate		Actual results match expected results	
2	Tests writeTiedWinnerCandidates	audit.writeTiedWinnerCandidate		Actual results match expected results	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_AuditTests_1.0

Name(s) of Testers: Hady

Test Description: Testing if the IR_Audit loser and winner candidate functionality works

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

file: IR_AuditTests.java,
method: test_IR_Audit_LosingCandidate(),
test_IR_Audit_WinningCandidate()

Automated: no

Results: Pass

Preconditions for Test: create a new default IR Audit constructor, IR_Audit audit = new IR_Audit()

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests writeLoser()	audit.writeLoser()		Actual results match expected results	
2	Tests writeWinner()	audit.writeWinner()		Actual results match expected results	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: PartyTests_1.0

Name(s) of Testers: Andrew Guerra

Test Description: Testing if the Party class works with its getter and setter methods

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

file: PartyTests.java,
method: testParty()

Automated: yes

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Test getPartyName	party.getPartyName("Gopher")	"Gopher"	"Gopher"	
2	Tests getBallotCount()	party.getSeatsWon()	0	0	
3	Tests setPartyName()	party.setPartyName("Badger")	None	None	
4	Tests setNumCandidates()	party.setNumCandidates(1)	None	None	
5	Tests setBallotCount()	party.setBallotCount(1)	None	None	
6	Tests setCandidateList()	party.setCandidateList(new ArrayList<String>(Arrays.asList("Bucky")))	None	None	
7	Tests getSeatsWon()	party.getSeatsWon()	1	1	
8	Tests getNumCandidates()	party.getNumCandidates()	1	1	
9	Tests getInitialBallotCount()	party.getInitialBallotCount()	0	0	

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: ElectionTests_1.0

Name(s) of Testers: Andrew Guerra

Test Description: Testing if the Election class works with its getter and setter methods

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

file: ElectionTests.java,
method: testElection()

Automated: yes

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Test constructor	new Election(0, "IR", "IR-eleciton.csv", null)	None	None	
2	Tests getNumBallots	election.getNumBallots()	0	0	
3	Tests getTypeElection()	election.getTypeElection()	"IR"	"IR"	
4	Tests getFileName()	election.getFileName()	"IR-election.csv"	"IR-election.csv"	
5	Tests getElectionFile()	election.getElectionFile()	null	null	
6	Tests setNumBallots()	election.setNumBallots(1)	None	None	
7	Tests setTypeElection()	election.setTypeElection("CPL")	None	None	
8	Tests setFileName()	election.setFileName("CPL-election.csv")	None	None	
9	Tests setElectionFile()	election.setElectionFile(null)	None	None	

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: CPL_Ballot_Test_1.0

Name(s) of Testers: Evan

Test Description: Testing the functionality of the
CPL_Ballot

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

Automated: no

file: CPL_BallotTests.java, methods: testCPL_Ballot

Results: Pass

Preconditions for Test: create a new CPL ballot object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getPartyNum	ballot.getPartyNum	1	1	
2	Tests getBallotNum	ballot.getBallotNum	2	2	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: Election_Test_1.0

Name(s) of Testers: Evan

Test Description: Test the functionality of the election class

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

file: ElectionTest.java, method: testElection

Automated: yes

Results: Pass

Preconditions for Test: create a new election

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getNumBallots	election.getNumBallots()	1	1	
2	Tests getTypeElection	election.getTypeElection	“CPL”	“CPL”	
3	Tests getFileName	election.getFileName()	“CPL-election.csv”	“CPL-election.csv”	
4	Tests getElectionFile	election.getElectionFile()	null	null	

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: PartyTest_1.0

Name(s) of Testers: Evan

Test Description: Testing if the party class functionality works as expected

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

Automated: yes

file: PartyTest.java, methods:

Results: Pass

Preconditions for Test: create a new party object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getPartyName	party.getPartyName	Badger	Badger	
2	Tests getBallotCount	party.getBallotCount	1	1	
3	Tests getSeatsWon	party.getSeatsWon	1	1	
4	Tests getNumCandidates	party.getNumCandidates	1	1	
5	Tests getInitialBallotCount	party.getInitialBallotCount	0	0	

Post condition(s) for Test:

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: CPL_Election_Tests_1.0

Name(s) of Testers: Evan

Test Description: Testing the full functionality of the CPL_Election class.

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

file: CPL_ElectionTests.java,
methods: testCPL_ElectionConstructor,
testCPL_ElectionReadCPLHeader,
testCPL_ElectionCandidateArray, testShuffleBallots,
testCPL_ElectionReadCPLBallots, testSetFirstSeatsAllocation,
testRemainderSeatsAllocation,

Automated: yes

Results: Pass

Preconditions for Test:

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Tests getAuditObject	test.getAuditObject	null	null	
2	Tests getElectionFile	test.getElectionFile	electionFile	electionFile	
3	Tests getNumParties	test.getNumParties	1	1	
4	Tests getParties	test.getParties	!null	!null	
5	Tests getNumSeats	test.getNumSeats	1	1	
6	Tests getNumBallots	test.getNumBallots	1	1	
7	Tests getCandidateList	test.getCandidatesList	candidate1	candidate1	

8	Tests getBallots	test.getBallots	!null	!null	
9	Tests getBallotNum	test.getBallotNum	0	0	
10	Tests getBallots after shuffle	test.getBallots	ballots order is not the same after shuffle	ballots order is not the same after shuffle	
11	Tests firstSeatAllocation	test.firstSeatAllocation		same as expected results	manual test
12	Tests remainderSeatAllocation	test.remainderSeatAllocation		same as expected results	manual test
13					
14					
15					

Post condition(s) for Test:

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: CPL_Election_Remainder_Tie

Name(s) of Testers: Andrew Guerra

Test Description: Manual testing to verify that coin flip randomly allocates a party with a seat given they have same amount of remaining votes

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

file: CPL_remainder_tie.csv

Automated: no

Results: Pass

Preconditions for Test: CPL_remainder_tie.csv exists in resources folder

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run election with CPL_remainder_tie.csv	CPL_remainder_tie.csv election data	valid CPL audit file	valid CPL file	
3	Run election with CPL_remainder_tie.csv until different winner in coin toss for step #1	CPL_remainder_tie.csv election data	valid CPL audit file that differs from result in step #1	valid CPL audit file that differs from result in step #1	

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: CPL_Election_Not_Enough_Candidates

Name(s) of Testers: Andrew Guerra

Test Description: Manual testing to verify that a party cannot be allocated more seats than they have candidates

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

file: CPL_landslide.csv

Automated: no

Results: Pass

Preconditions for Test: CPL_remainder_tie.csv exists in resources folder

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run election with CPL_landslide.csv	CPL_landslide.csv election data	valid CPL audit file with parties not having more seats than candidates	valid CPL audit file with parties not having more seats than candidates	

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: System

Test Date: March 26, 2023

Test Case ID#: IR_Tie

Name(s) of Testers: Andrew Guerra

Test Description: Manual testing to verify that when a tie occurs, a random party is chosen to be removed and their ballots reallocated

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

file: IR_tie.csv

Automated: no

Results: Pass

Preconditions for Test: IR_tie.csv exists in resources folder

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run election with IR_tie.csv	IR_tie.csv election data	valid IR audit file with a tie with one candidate removed randomly	valid IR audit file with a tie with one candidate removed randomly	
3	Run election with IR_tie.csv until different lose in tie	IR_tie.csv election data	valid IR audit file with a tie with different candidate removed randomly	valid IR audit file with a tie with different candidate removed randomly	

Post condition(s) for Test: None

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_Election_UnitTest 1.0

Name(s) of Testers: Hady Kotifani

Test Description: Testing constructors, getters and setters

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

file: CPL_ElectionTests.java, methods: getCandidates, getNumCandidates(), getNumRemainingCandidates(), getCurrentBallots(), getCurrentBallotCount(), getAudit(), setCandidates, setNumCandidates, setNumRemainingCandidates, setCurrentBallots, setCurrentBallotCount, setAudit, IR_Election(electionFile, date) constructor

Automated: yes

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
--------	-----------------------	-----------	-----------------	---------------	-------

1	call the constructor with electionFile , date				
2	getCandidates	ir.getCandidates	null	null	
3	getNumCandidates	ir.getNumCandidates	0	0	
4	getCurrentBallots	ir.getCurrentBallots	null	null	
5	getCurrentBallotCount	ir.getCurrentBallotCount	0	0	
6	getAudit	ir.getAudit	not null	not null	
7	getNumRemainingCandidates	ir.getNumRemainingCandidates	0	0	
8	Set all values of ir election	ir.getCandidates	not null	not null	
9		ir.getNumCandidates	1	1	
10		ir.getCurrentBallots	not null	not null	
11		ir.getCurrentBallotCount	1	1	
12		ir.getAudit	null	null	
13		ir.getNumRemainingCandidates	1		

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_Election_UnitTest 1.1

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election ties

Automated: No	Indicate where are you storing the tests (what file) and the name of the method/functions being used. file: IR_Election_Tests.java, ir.run()
Results: Pass	

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	inititalize ir election				
2	send ir election tie.csv file		Each candidate wins once through multiple runs of test, audit displays candidate names who tied with lowest votes	Each candidate wins once through multiple runs of test, audit displays candidate names who tied with lowest votes	
3	call run on ir	tie_ir.csv	Each candidate wins once through multiple runs of test, audit displays candidate names who tied with lowest votes	Each candidate wins once through multiple runs of test, audit displays candidate names who tied with lowest votes	

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_Election_UnitTest 1.3

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election shuffles

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	inititalize ir election				
2	send ir election shuffle.csv file	shuffle.csv			
3	call run on ir	shuffle.csv	The order in which ballot IDs are printed should be different each run in the audit	The ballot IDs are in a different order each time ir is run	

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_Election_UnitTest 1.4

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election win by majority

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	inititalize ir election				
2	send ir election csv file	majority.csv			
3	call run on ir		The candidate with over 50% of votes should win	The candidate with over 50% votes does win	

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: IR_Election_UnitTest 1.5

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election wins by plurality

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	inititalize ir election				
2	send ir election csv file	plurality.csv			
3	call run on ir		The candidate with less than 50% of votes but more than the other remaining candidates should win	The candidate with less than 50% of votes but more than the other remaining candidates does win	

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System**Team# 21****Test Stage:** Unit**Test Date:** March 26, 2023**Test Case ID#:** IR_Election_UnitTest 1.6**Name(s) of Testers:** Hady Kotifani**Test Description:** Testing ir election ballot reallocations**Indicate where are you storing the tests (what file) and the name of the method/functions being used.****Automated:** No

file: IR_Election_Tests.java, ir.run()

Results: Pass**Preconditions for Test:** None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	allocation.csv			
3	call run on ir		Ballots from removed candidates should either be tossed out or reallocated to the candidate of the ballots next rank	Ballots from removed candidates is either be tossed out or reallocated to the candidate of the ballots next rank	

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 27, 2023

Test Case ID#: IR_Election_UnitTest 1.7

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election advanced ballot reallocations

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

file: IR_Election_Tests.java, ir.run()

Automated: No

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	allocationBallotAdvanced.csv			
3	call run on ir		Ballots from removed candidates should either be tossed out or reallocated to the candidate of the ballots next rank,	Ballots from removed candidates is either be tossed out or reallocated to the candidate of	

			ballot should update from rank 1 to ranking 3 even if the candidate at ranking 2 had been eliminated	the ballots next rank, and does update rank properly and is not thrown out	
--	--	--	--	--	--

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 27, 2023

Test Case ID#: IR_Election_UnitTest 1.8

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election ballot with double digit candidates

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	DoubleDigitCandidate.csv			

3	call run on ir		Ballots should be processed and be properly allocated for ten candidates	Ballots are properly processed and allocated to the ten candidates	
---	----------------	--	--	--	--

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 27, 2023

Test Case ID#: IR_Election_UnitTest 1.9

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election ballot reallocations for 100,000 ballots, stress test with ten candidates

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	allocation.csv			

3	call run on ir		allocate 100,000 ballots, with ten candidates in under four minutes	Did allocate 100,000 ballots in under four minutes	
---	----------------	--	---	--	--

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 28, 2023

Test Case ID#: IR_Election_UnitTest 1.10

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election ballot with only one candidate

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	oneCandidate.csv			

3	call run on ir		No crashing and candidate wins	Did not crash and candidate won properly	
---	----------------	--	--------------------------------	--	--

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 28, 2023

Test Case ID#: IR_Election_UnitTest 1.11

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election works with only two candidates with long party names and names

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
file: IR_Election_Tests.java, ir.run()

Automated: No

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	twoCandidateLongname.csv			
3	call run on ir		Candidate party and name should be displayed properly	Names are displayed properly and	

			and proper candidate should win through plurality or majority	correct corresponding candidate	
--	--	--	---	---------------------------------	--

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 28, 2023

Test Case ID#: IR_Election_UnitTest 1.12

Name(s) of Testers: Hady Kotifani

Test Description: Testing ir election works and disregards ballots that don't have enough rankings for candidates still in the running

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

Automated: No

file: IR_Election_Tests.java, ir.run()

Results: Pass

Preconditions for Test: None

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize ir election				
2	send ir election csv file	throwOutBallot.csv			
3	call run on ir		Ballots #4-#9 should be discarded and the rest kept	Ballots #4-#9 are disregarded	

				and the rest are kept	
--	--	--	--	-----------------------	--

Post condition(s) for Test: N/A

Project Name: Project 1: Voting System

Team# 21

Test Stage: Unit

Test Date: March 26, 2023

Test Case ID#: CPL_AuditTests.java

Name(s) of Testers: Evan

Test Description: Testing the functionality of CPL Audit

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

file: CPL_AuditTests.java, methods: testCPL_AuditConstructor, testCPL_AuditClose, testCPL_AuditGettersAndSetters, testCPL_AuditWriteRemainingVotes,

Automated: no

Results: Pass

Preconditions for Test:

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Test getDate	test.getDate()	date	date	
2	Tests getAuditFile	test.getAuditFile	!null	!null	
3	Tests getAuditWriter	test.getAuditWriter	!null	!null	
4	Tests setDate	test.setDate	N/A	N/A	

5	Tests setCandidateList	test.setCandidateList	N/A	N/A	
6	Tests setNumCandidate	test.setNumCandidate	N/A	N/A	
7	Tests writeHeaderToFile	test.writeHeaderToFile	actual results matched expected	actual results matched expected	
8	Tests setInitialBallotCount	test.setInitialBallotCount	N/A	N/A	
9	Tests writeInitalPartyVotes	test.writeInitialPartyVotes	actual results matched expected	actual results matched expected	
10	Tests writeFirstSeatsAllocation	test.writeFirstSeatsAllocation	actual results matched expected	actual results matched expected	
11	Tests setInitialBallotCount	test.setInitialBallotCount	N/A	N/A	
12	Tests writeRemainingVotes	test.writeRemainingVotes	actual results matched expected	actual results matched expected	
13	Tests addBallot	test.addBallot	actual results matched expected	actual results matched expected	
14	Tests writeGreatestRemainder Party	test.writeGreatestRemainderParty	actual results matched expected	actual results matched expected	
15	Tests extraSeats	test.extraSeats	3	3	
16	Tests incSeatsWon	test.incSeatsWon	1	1	

Post condition(s) for Test: