Test logs:

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: C01
Test Description: We create different instances of Candidate class and see if their constructors work well
Name(s) of Testers: Yuanzong Zhang & Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Candidate Class)
Automated: no
Results: Pass

Preconditions for Test:

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	construct a candidate vector	vector<Candidate> candidates;			No result
2	Instantiate three candidates with different name, and put them into candidate vector	Candidate c1 ("x"); Candidate c2 ("y"); Candidate c3 ("z");			No result
3	Check the candidate name	x, y, z	Candidates names: x y z	Candidates names: x y z	

Post condition(s) for Test: Each candidate can add Ballot and return their number of votes

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: C02
Test Description: We add a ballot to a candidate's ballots vector and see if it got added successfully
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test addBallot)
Automated: no
Results: Pass

Preconditions for Test: Ballot class instance has already been created and has preference map in it.

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Construct an instance of Candidate class	Candidate c_test("Test addBallot");			No result
2	Call addBallot() function	c_test.addBallot(b 2);			No result
3	Check the ballot added	1,2,3,4,5,a,b,c,d,e	Ballot 2: 1 a 2 b 3 c 4 d 5 e	Ballot 2: 1 a 2 b 3 c 4 d 5 e	

Post condition(s) for Test: ballot has been successfully assigned to a specific candidate, it can later be used to count a candidate's total votes or generate report.

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: B01
Test Description: We create different instances of Ballot class and see if their constructors work well
Name(s) of Testers: Yuanzong Zhang & Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Class Ballot)
Automated: no
Results: Pass

Preconditions for Test:

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	construct a candidate ballot	vector<Ballot> ballots;			No result
2	Instantiate a Ballot	Ballot b1(map1, 1); Ballot b2(map2, 1);			No result

3	Pushback to vector ballot:	ballots.push_back(b1);
		
		ballots.push_back(b2);
					No result

4	Add candidate name just as the first test case , and their preference to the Ballot	1,2,3,x,y,z 1,2,3,4,5,a,b,c,d,e	Ballot 1: 1 x 2 y 3 z Ballot 2: 1 a 2 b 3 c 4 d 5 e	Ballot 1: 1 x 2 y 3 z Ballot 2: 1 a 2 b 3 c 4 d 5 e	

Post condition(s) for Test: This Ballot can view the candidates' name and their preference in a map

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: B02
Test Description: We create different instances of Ballot class and see if their get_id() function work well
Name(s) of Testers: Yuanzong Zhang & Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp
Automated: no
Results: Pass

Preconditions for Test: Already construct a ballot vector

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	construct a candidate ballot	vector<Ballot> ballots;			No result
2	Instantiate a Ballot	Ballot b1(map1, 1); Ballot b2(map2, 2);			No result

3	Pushback to vector ballot:	ballots.push_back(b1);
		
		ballots.push_back(b2);
					No result

4	Add candidate info just as the first test case , and their preference to the Ballot				
5	Check id	1, 2	Ballot ID(inside): 1 Ballot ID(inside): 2	Ballot ID(inside): 1 Ballot ID(inside): 2	

Post condition(s) for Test: Ballot Id can be used when we generate report

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: P01
Test Description: By passing different parameters and to see if Assign_ballot works well
Name(s) of Testers: Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp, assign_ballot()
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality, and initiate a plurality object

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call assign_ballot ()	p.assign_ballot();			
3	Check the result print on the termial	temp = p.get_candidates_array(); Print information for temp[0] and temp[1]	x, 3 y, 1	x, 3 y, 1	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: P02
Test Description: By passing different parameters into the plurality constructor and to see if generate_result works well
Name(s) of Testers: Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp generate_result()
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call generate_result()	p.generate_result();			
3	Get winner list	vector<Candidate> winners = result.get_winner_list();			
4	Get loser list	vector<Candidate> losers = result.get_loser_list();			
5	Check the winner and looser		6 winners the x:3 a:2 b:2 y:1	the result is that 6 win the voting they are : x:3 a:2	

			z:1 C:1 2 loosers d:0 e:0	b:2 y:1 z:1 C:1 the result is that 2 loose the voting they are : d:0 e:0	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning, and we write the detailed information into a “report.txt” file, and print the winner names on the screen.

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: S01
Test Description: By passing different parameters and to see if Assign_ballots works well
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality, and initiate a STV object

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			
2	Call assign_ballot ()	p.assign_ballot();			
3	Check the result print on the termial	temp = p.get_candidates_array(); Print information for temp[0] and temp[1]	x, 3 y, 1	x, 3 y, 1	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: S02
Test Description: By passing different parameters into the constructor and to see if the generate_report function works well
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call generate_result()	p.generate_result();			
3	Get winner list	vector<Candidate> winners = result.get_winner_list();			
4	Get loser list	vector<Candidate> losers = result.get_loser_list();			
5	Check the winner and looser		6 winners the x:3 a:2 b:2 y:1 z:1	the result is that 6 win the voting they are : x:3 a:2 b:2	

			C:1 2 loosers d:0 e:0	y:1 z:1 C:1 the result is that 2 loose the voting they are : d:0 e:0	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning, and we write the detailed information into a “report.txt” file, and print the winner names on the screen.

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: I01
Test Description: We created instance of Input class and see if its constructor works well
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Input Class)
Automated: no
Results: Pass

Preconditions for Test: The main function got runned and user has input needed information

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Construct an instance of Input class	Input input;			No result

2	Using setter functions in Input class to set values	input.set_num_seats(6);
		
		input.set_num_ballots(10);
		
		input.set_num_candidates(8);
		
		input.set_algorithm(1);
					No result

3	Check the value stored in Input class	10, 8, 6, 1	Number of ballots: 10 Number of candidates: 8 Number of seats: 6 Choose of algorithm: 1	Number of ballots: 10 Number of candidates: 8 Number of seats: 6 Choose of algorithm: 1	

Post condition(s) for Test: store user input information in Input class and can be passed to Plurality or STV class to help processing voting data.

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: R01
Test Description: We created instance of Result class and see if it successfully stored the result generated by Plurality or STV
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Result Class)
Automated: no
Results: Pass

Preconditions for Test: The program has finished processing all voting data

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes

1	Create two vectors of Candidate to store winners and losers	vector<Candidate > winners = result.get_winner_ list();
		
		vector<Candidate > winners = result.get_loser_lis t();
					No result

2	Get all ballots assigned to winners	vector<Ballot> ballot_report = winners[i].get_ball ots();			No result
3	Check winner list	x: 1,3,4 a: 2,5 b: 7,9 y: 10 z: 6 c: 8	x: 1,3,4 a: 2,5 b: 7,9 y: 10 z: 6 c: 8	x: 1,3,4 a: 2,5 b: 7,9 y: 10 z: 6 c: 8	
4	Check loser list	d, e	d, e	d, e	

Post condition(s) for Test: None

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File01
Test Description: In this test, 8 voters will vote 4 seats out of 11 candidates with Plurality and STV method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile1.csv
Automated: no
Results: Pass

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile1 .csv			
3	User input the number of seat 4				
4	User input “1” to choose plurality method		Winner should be any four candidates of A, B, C, D, G, K, J and L. Looser should be the rest four of winner list and E, F, H and I.	Plurality Results: Winner of Seat 1: [A] Winner of Seat 2: [B] Winner of Seat 3: [C] Winner of Seat 4: [D]	The winner is four random candidates from A, B, C, D ,G, K, J and L.
5	User input “2” to choose STV method and choose not to shuffle		Winner should be C, J, and B And because we only have 3 winners and A	STV (Droop Quota) Results: Winner of Seat 1: [C]	

			is the ‘best’ loser, so A got chosen when we asked for 4 winners	Winner of Seat 2: [J] Winner of Seat 3: [B] Winner of Seat 4: [A] (did not meet droop)	
6	User input “2” to choose STV method and choose to shuffle		Winner should be C, J, and B And because we only have 3 winners and A is the ‘best’ loser, so A got chosen when we asked for 4 winners	STV (Droop Quota) Results: Winner of Seat 1: [C] Winner of Seat 2: [J] Winner of Seat 3: [B] Winner of Seat 4: [A] (did not meet droop)	‘To shuffle’ and ‘not to shuffle’ have same result.

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File02
Test Description: In this test, 11 voters will vote 1 seat out of 5 candidates with Plurality method and STV method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile2.csv
Automated: no
Results: Pass
Preconditions for Test:None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile2.csv			
3	User input the number of seat 1				
4	User input “1” to choose plurality method		Winner should be any one candidats of C and D.	Plurality Results: Winner of Seat 1: [C]	The winner is one random candidates out of C and D.
5	User input “2” to choose STV method and choose not to shuffle		The winner should be C	STV (Droop Quota) Results: Winner of Seat 1: [C]	
6	User input “2” to choose STV method and choose to shuffle		The winner should be C	STV (Droop Quota) Results: Winner of Seat 1: [C]	‘To shuffle’ and ‘not to shuffle’ have same result.

Post condition(s) for Test:Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File03
Test Description: In this test, 22 voters will vote 2 seats out of 15 candidates with Plurality method and STV method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile3.csv
Automated: no
Results: Pass

Preconditions for Test:None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile3.csv			
3	User input the number of seat 2				
4	User input “1” to choose plurality method		Winner should be D and F. Looser should be the rest 13 candidates.	Plurality Results: Winner of Seat 1: [D] Winner of Seat 2: [F]	
5	User input “2” to choose STV method and choose not to shuffle		The winner should be L and D.	STV (Droop Quota) Results: Winner of Seat 1: [L] Winner of Seat 2: [D]	
6	User input “2” to choose STV method and choose to shuffle		The winner should be L and D.	STV (Droop Quota) Results: Winner of Seat	‘To shuffle’ and ‘not to shuffle’ have same result.

				1: [L] Winner of Seat 2: [D]	

Post condition(s) for Test:Winner & Loser information are printed to the screen and a report containing ballots' assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File04
Test Description: In this test, there are two candidates with same name:"[B]", and we run both Plurality and STV
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile4\(duplicative\ candidate\ name\).csv
Automated: no
Results: Fail

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile4\(duplicative\ candidate\ name\).csv			
3	User input the number of seat 2				
4	User input "1" to choose plurality method		Failed	Plurality Results: Winner of Seat 1: [D] Winner of Seat 2: [H]	The system does not report error. This bug is shown in "Buglist".
5	User input "2" to choose STV method and choose not to shuffle		Failed	STV (Droop Quota) Results: Winner of Seat 1: [D] Winner of Seat 2: [H]	The system does not report error. This bug is shown in "Buglist".
6	User input "2"		Failed	STV (Droop	The system

	to choose STV method and choose to shuffle			Quota) Results: Winner of Seat 1: [H] Winner of Seat 2: [D]	does not report error. This bug is shown in “Buglist”.

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File05
Test Description: In this test, second voters gives two primary ballot to two candidates. We test with Plurality and STV method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile5\(duplicative\ ballot\).csv
Automated: no
Results: Fail

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile5\(duplicative\ ballot\).csv	Failed	System does not report error.	This bug is list in “Buglist”.
3	User input the number of seat 3				
4	User input “1” to choose plurality method		Failed	Plurality Results: Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: [H]	The system does not report error. This bug is shown in “Buglist”.
5	User input “2” to choose STV method and choose not to shuffle		Failed	STV (Droop Quota) Results: Winner of Seat 1: [D]	The system does not report error. This bug is shown in “Buglist”.

				Winner of Seat 2: [B] Winner of Seat 3: [H] (did not meet droop)	
6	User input “2” to choose STV method and choose to shuffle		Failed	STV (Droop Quota) Results: Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: [H] (did not meet droop)	The system does not report error. This bug is shown in “Buglist”.

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File06
Test Description: In this test, test file is blank file with 3 seat being chosen.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile6\(blank\).csv
Automated: no
Results: Pass

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile6\(blank\).csv		.	
3	User input the number of seat 3		Program terminated	Error! More seats than candidates. Now exiting..	

Post condition(s) for Test: Program reports error and terminates.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File07
Test Description: In this test, each voter has 10 votes, but there is only 6 candidate name.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: ./tests/examplefile7\(missing candidate name\).csv
Automated: no
Results: Fail

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile7\(missing candidate name\).csv	Failed		
3	User input the number of seat 5				
4	User input “1” to choose plurality method		Failed	Plurality Results: Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: Winner of Seat 4: Winner of Seat 5:	The system does not report error. This bug is shown in “Buglist”.
5	User input “2” to choose		Failed	STV (Droop Quota) Results:	The system does not

	STV method and choose not to shuffle			Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: Winner of Seat 4: Winner of Seat 5: [A] (did not meet droop)	report error. This bug is shown in “Buglist”.
6	User input “2” to choose STV method and choose to shuffle		Failed	STV (Droop Quota) Results: Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: Winner of Seat 4: [F] Winner of Seat 5: [A] (did not meet droop)	The system does not report error. This bug is shown in “Buglist”.

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File08
Test Description: In this test, the CSV file has invalid ballot -- the preference number for some voters is larger than the number of candidates.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile8\(\(invalid\ ballot\).csv

Automated: no
Results: Fail
Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile8\(\(invalid\ ballot\).csv	Failed	System does not report error.	This bug is list in “Buglist”.
3	User input the number of seat 3				
4	User input “1” to choose plurality method		Failed	Plurality Results: Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: [A]	The system does not report error. This bug is shown in “Buglist”.
5	User input “2” to choose STV method and choose		Failed	Winner of Seat 1: [D] Winner of Seat 2:	The system does not report error. This bug is

	not to shuffle			[B] Winner of Seat 3: [F] (did not meet droop)	shown in “Buglist”.
6	User input “2” to choose STV method and choose to shuffle		Failed	Winner of Seat 1: [D] Winner of Seat 2: [B] Winner of Seat 3: [A] (did not meet droop)	The system does not report error. This bug is shown in “Buglist”.

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File09
Test Description: In this test, we try to test different candidate name length with Plurality method and STV method. For example using [AB] , rather than A.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile9(invalid candidates' name).csv
Automated: no
Results: Pass
Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile9(invalid candidates' name).csv			
3	User input the number of seat 5				

4	User input “1” to choose plurality method		Winners are must include D and E, F, but the other two are random	Plurality Results: Winner of Seat 1: [D] Winner of Seat 2: [E] Winner of Seat 3: [F] Winner of Seat 4: [AB] Winner of Seat 5: [B]	
5	User input “2” to choose STV method and choose not to shuffle		Winners are D, E, C, F, AB.	STV (Droop Quota) Results: Winner of Seat 1: [D] Winner of Seat 2: [E] Winner of Seat 3: [C] Winner of Seat 4: [F] Winner of Seat 5: [AB] (did not meet droop)	
6	User input “2” to choose STV method and choose to shuffle		Winners are F, D, B, E, AB.	STV (Droop Quota) Results: Winner of Seat 1: [F] Winner of	

				Seat 2: [D] Winner of Seat 3: [B] Winner of Seat 4: [E] Winner of Seat 5: [AB] (did not meet droop)	

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots' assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: April 3
Test Case ID#: File10
Test Description: In this test, we try to test the original csv file provided by Shana.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: /tests/examplefile.csv

Automated: no
Results: Pass
Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile.csv			
3	User input the number of seat 5				
4	User input “1” to choose plurality method		A and G are guaranteed to be chosen as a Winner. And the other 3 should be randomly chosen from the rest candidates.	Plurality Results: Winner of Seat 1: [A] Winner of Seat 2: [G] Winner of Seat 3: [B] Winner of Seat 4: [C] Winner of Seat 5: [D]	

5	User input “2” to choose STV method and choose not to shuffle		Winners are A, G, E, B and C.	STV (Droop Quota) Results: Winner of Seat 1: [A] Winner of Seat 2: [G] Winner of Seat 3: [E] Winner of Seat 4: [B] Winner of Seat 5: [C]	
6	User input “2” to choose STV method and choose to shuffle		Winners are G, A, D, B and C.	STV (Droop Quota) Results: Winner of Seat 1: [G] Winner of Seat 2: [A] Winner of Seat 3: [D] Winner of Seat 4: [B] Winner of Seat 5: [C]	

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.