Project Name: Project 1: Voting System	Team# 12
Test Stage: Unit X System	Test Date: Nov.16 / 2019
Test Case ID#: 001 Test Description: "random_int" method: Test random integer generator when for integers list, which is called "Flipcoin" in our code to make sure it is a fair call.	Name(s) of Testers: Yingjin Zhang, Sunny Qin
Automotod: vos no V	Indicate where are you storing the tests (what file) and the name of the method/functions being used. ./Test/Test_flipcoin_int.java
Automated: yes no X	
Results: Pass X Fail	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
	Test Flip_coins for 2 integers Run random generator 1000 times	Test_flipcoin_int.java	Times of 1: around 500 Times of 2: around 500	Times of 2: 501	The result is not the exact number, but in an acceptable range.
	Test Flip_coins for more than 2 integers Run random generator 1000 times	Test_flipcoin_int.java	Times of 1: around 333 Times of 2: around 333 Times of 3: around 333	Times of 2: 342	The result is not the exact number, but in an acceptable range.
3					

Preconditions for Test: A tie appears in ballots of parties or candidates when allocating seats.

Post condition(s) for Test:

The random generator for integers is a fair call, which means that nearly equal probability for the two elements in the list .

Project Name: The project #, name of your system, and the team#

Test Stage: Indicate whether it is a unit test or a system test.

Test Date: The date the test was performed.

Test Case ID#: A unique ID is required. Decide on a naming convention and use numbering. Example: Ballot_Shuffle_1

Name(s) of Testers: List the names of anyone involved in running this test case.

Test Description: Describe briefly the test objective.

Automated: Indicate if the test is completely automated or being checked manually. (If you have methods running the tests and checking results, select "yes". If you are manually checking results, indicate manual by selecting the "no.")

Results: Indicate if the test passed or failed.

Step #: You will be listing the test steps in order. This number is the step number in the process.

Test Step Description: Details of the test step.

Test Data: What the test data will be for this step. Be clear on what the input data will be. If using a specific file, be clear on the name.

Expected Result: What result are you expecting from the program component or system.

Actual Result: What result were returned based on the test.

Post condition for Test: What will be true after the test has been run? Has the state of the system changed in any way?

Notes: Comments and notes for you and your team members.

Project Name: Project 1: Voting System Team# 12 Test Stage: Unit X System ___ **Test Date: Nov.16/2019** Test Case ID#: 002 Name(s) of Testers: Yingjin Zhang, Sunny Qin **Test Description:** "random str" method: Test random integer generator when for a string list, which is called "Flipcoin" in our code to make sure it is a fair call. Indicate where are you storing the tests (what file) and the name of the method/functions being used. Automated: yes_ no X ./Test/Test_flipcoin_str.java Results: Pass X Fail

Preconditions for Test: A tie(same ballots) appears when generate results for candidates.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
	Test flipcoin for two candidates with same votes Run random generator 1000 times	Test_flipcoin_str.java	Times of 'a': around 500 Times of 'b': around 500	Times of 'b': 508	The result is not the exact same number, but in an acceptable range.
	Test flipcoin for three more candidates with same votes Run random generator 1000 times	Test_flipcoin_str.java	Times of 'a': around 333 Times of 'b': around 333 Times of 'c': around 333	Times of 'b': 317	The result is not the exact same number, but in an acceptable range.

Post	condition	(0)	for	Test.
Post	conamon	S) IOI	rest:

The random generator for string is a fair call, which means that nearly equal probability for each element in the list.

Project Name: Project 1: Voting System	Team# 12
Test Stage: Unit X System	Test Date: Nov. 16 2019
Test Case ID#: 003 Test Description: "allocate_seat" method: Test allocating seat method to see whether it can allocate correct seats to each party.	Name(s) of Testers: Yingjin Zhang, Sunny Qin
Automated: yes no X	Indicate where are you storing the tests (what file) and the name of the method/functions being used. ./Test/Test_process_allocateseats.java

Preconditions for Test: The flipcoin method is well designed, the ballots for parties are parsed correctly, the number of seats has also been correctly read.

Results: Pass X

Fail_

Step	Test Step	Test	_ <u> </u>	Actual	
#	Description	Data	Result	Result	Notes
1		party 1: 100 votes party 2: 230 votes	party 2: 2	party 2: 2	The right side is the seats number assigned to the corresponding party.
2	occurs, need flip a coin to	Test_process_allocateseats.java: party 1: 100 votes party 2: 250 votes party 3: 350 votes	party 2: 2	party 2: 2	The result is random generated. It depends on the random generator result.

3	Test a tie between three parties occurs, need flip a coin to	Test_process_allocateseats.java: party 1: 270 votes	party 2: 3		The result is random generated. It depends on the random generator result.
	Test seats > cands case	party 1: 1 votes, 3 candidates	party 2: 2	μ ,	The right side is the seats number assigned to the corresponding party.
4		seats = 4			

Post condition(s) for Test:

The seats are correctly allocated to parties based on the seats allocation algorithm described in instructions.

Preconditions for Test: The OPL and CPL work well, and have already generated results.

Project Name: Project 1: Voting System	Team# 12
Test Stage: Unit X System	Test Date: Nov. 16 2019
Test Case ID#: 004 Test Description: "display_results" method: Testing displaying results method to see whether the voting results can be correctly printed to the screen.	Name(s) of Testers: Yingjin Zhang, Sunny Qin
Automated: yes no X	Indicate where are you storing the tests (what file) and the name of the method/functions being used. /Test/Test_process_display.java
Results: Pass X Fail	

Step #	Test Step Description	Test Data	Expected Result			Actual Result			Notes
			Candidate	Darty I	Ballots / Rank	Candidate	Party Ra	llots / Rank	In OPL, the last column
			b	D	60	b	D Taily Ba	60	shows the ballots for each
			a	D	40	a	D	40	candidate.
			e e	G	110	e e	G	110	candidate.
			d	G	90	d	G	90	
			C	G	50	C	G	50	
			l i	Ī	20	i	Ī	20	
			i	Ī	90	i	Ī	90	
		Test_process_display.	h	Ī	110	h	Ī	110	
		java	g	I	80	g	I	80	
1	Test display for OPL results		f	I	50	f	I	50	
			Candidate	Party E	Ballots / Rank	Candidate	Party Ba	allots / Rank	In CPL, the last column shows
			b	Ď	2	b	Ď	2	the rank for each candidate.
			a	D	1	a	D	1	
			e	G	3	e	G	3	
			d	G	1	d	G	1	
			С	G	2	c	G	2	
			j	I	3	j	I	3	
			i	I	2	i	I	2	
			h	I	1	h	I	1	
		Test_process_display.j		I	5	g	I	5	
2	Test display for CPL results	ava	f	I	4	f	I	4	

Post condition(s) for Test:

Display the results in a table and correctly print it in the terminal.

Team# 12

Test Stage:	Unit X	System	Test Date: Nov. 16 2019

Test Case ID#: 005 Name(s) of Testers: Yingjin Zhang, Sunny Qin

Test Description:

"generate result CPL" method:

Test generate results method for CPL to see whether it can generate correct results based on candidates' votes and number of seats.

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

./Test/Test cpl generateResult.java

Automated: yes___ no X

Results: Pass X Fail_____

Preconditions for Test: Allocates_seats, flipcoin, and parserCPL methods work well.

Step	Test Step	Test	Expected	Actual	Notes
#	Description	Data	Result	Result	Notes
			a : 1 b : null	a:1 b:null	cadidates_ballots contains each candidate and his/her
	Test regular CPL with regular		b : nun	b : nun	corresponding rank in the
	seats and votes	The regular part in	c:2	c:2	party.
		Test_cpl_generateResult.jav	d:1	d :1	The string on the left of the
		a	e : null	e : nul	column is the name of the
		party_seats[1,2,3]		1	candidate. The number on the
		candidats_ballots[]	f :null	f :null	right of the column is the rank
			g : null	g : null	of the corresponding
			h:1	h:1	candidats. a 'null' represents
1			i:2	i : 2	that the candidate is not been
1			j:3	j:3	elected.
	Test CPL results with 0 seat	Test_cpl_generateResult.java	a : null	a : null	cadidates_ballots contains
	assigned for a party	party_seats[0,2,3]	b : null	b : nul	each candidate and his/her
		candidats_ballots[]			corresponding rank in the
			c : 2	c:2	party.
			d : 1	d:1	The string on the left of the
2			e : null	e : null	column is the name of the

	f: null g: null h: 1 i: 2 j: 3	f: null g: null h: 1 i: 2 j: 3	candidate. The number on the right of the column is the rank of the corresponding candidats. a 'null' represents that the candidate is not been elected.

Post	condition	(\mathbf{z})	for	Test:
1 031	Continuon	01	101	1030

The voting results are generated and the candidates are correctly picked based on the rank of candidates and number of seats for parties.

Project N	lame: Project 1: Voting System		Team# 12
TF 4 C4	TT 1: TT	T	

Test Stage: Unit X System __ Test Date: Nov. 16 2019

Test Case ID#: 006 Name(s) of Testers: Yingjin Zhang, Sunny Qin

Test Description:

"generate result OPL" method:

Test generate results method for OPL to see whether it can generate correct results based on candidates' votes and number of seats.

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_opl_generateResult.java

Automated: yes___ no X

Results: Pass X Fail_____

Preconditions for Test: Allocates_seats, flipcoin, and parserOPL methods work well.

Step #	Test Step Description	Test Data	•	Actual Result	Notes

	_				
			a : null	a : null	The string on the left of the
	Test regular generate seats with		b : 200	b : 200	column is the name of the
	regular data, no AC or EX	The regular part in	0.200	0.200	candidate. The number on the
		Test_opl_generateResult.jav	220	220	
		a	c : 220	c : 220	right of the column is the
			d : null	d : null	number of ballots of the
		party_seats[1,2,3]	e: 330	e : 330	corresponding candidats. a
		candidats_ballots[]			'null' represents that the
			f: 410	f: 410	candidate is not been elected.
			g: 520	g:520	
			h : null	h : null	
			i : null	i : null	
1			j: 330	j: 330	
1			y. 220	j . 220	
	Test OPL results with 0 seat		a : null	a : null	The string on the left of the
I	assigned for a party		b : null	b : null	column is the name of the
	assigned for a party	Test_opl_generateResult.jav	o . nun	U. Hull	
		a:	200	200	candidate. The number on the
			c : 200	c : 200	right of the column is the
		party_seats[0,2,3]	d : null	d : null	number of ballots of the
		candidats_ballots[]	e: 330	e : 330	corresponding candidats. a
		candidats_banots[]			'null' represents that the
			f: 440	f: 440	candidate is not been elected.
			g:510	g:510	
			h : null	h : null	
			i : null	i : null	
2			j: 330	j: 330	
			j. 550	J. 330	
			a : null	a : null	The string on the left of the
		Test_opl_generateResult.java	b:100	b:100	column is the name of the
		·			candidate. The number on the
	Test tie occurred between two		c : 200	c : 200	right of the column is the
	candidates with the same votes		d : null	d : null	number of ballots of the
		party_seats[1,2,3]	e: 330	e: 330	corresponding candidats. a
			c . 330	c . 330	'null' represents that the
		candidats_ballots[]	f: 440	f: 440	candidate is not been elected.
					candidate is not been elected.
			g: 440	g: 440	TTI
			h : null	h : null	There are eight expected
			i : null	i : null	values since there is a tie in
			j:330	j: 330	each party, but they are hard to
					list, so we only list the output
3	l.				one.
	Tost tie ecourred between the		a : null	a : null	The string on the left of the
	Test tie occurred between three		1 150	b : 150	column is the name of the
	more candidates with the same	rest_opt_generateResuit.Java	2.150	0.150	candidate. The number on the
	votes.		c : null	c : null	right of the column is the
		party_seats[1,1,3]	d: 200	d : 200	number of ballots of the
		candidats_ballots[]			
			e : null	e : null	corresponding candidats. a
					'null' represents that the
			f : null	f : null	candidate is not been elected.
			g:440	g:440	
			h : null	h : null	There are nine expected
			i : 440	i : 440	values since there is a tie in
1			j:530	j: 530	each party, but they are hard to
4	l	1	j. 330	J . 550	party, out they are natu to

			list, so we only list the output one.

Post condition(s) for Test:

The voting results are generated and the candidates are correctly picked based on the ballots of candidates and number of seats for parties.

Project Name: Project 1: Voting System

Team# 12

Test Stage: Unit X System __ Test Date: Nov. 16 2019

Test Case ID#: 007 Name(s) of Testers: Xiaohui Chao

Test Description:

Test the "writeSummaryReport" method to see if the

election result is saved in the summary file

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_WriteSummary.java

Automated:	yes	no	X
-------------------	-----	----	---

Results: Pass X Fail____

Preconditions for Test: The seats are allocated and the winners are generated.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
		The regular part in Test WriteSummary.java	Election Summary Report	Election Summary Report	Write success
	Test write summary results to		The voting type is OPL.	The voting type is OPL.	
1	file for OPL	votingType = "OPL"			

porty Ballots = I Independent Candidates, Conductions 2, Germa-1. Republican-2, Democratic-3, Reform-1 Democratic-3, Reform-1 Party: Green, Number of Votes: 0 Party: Green, Number of Votes: 1 Party: Republican, Number of Votes: 1 Party: Republican, Number of Votes: 2 Party: Democratic, Number of Votes: 2 Party: Democratic, Number of Votes: 2 Party: Democratic, Number of Votes: 3 Party: Reform, Number of Votes: 1 Party: Permocratic, Number of Votes: 3 Party: Permocratic, Candidates: Numy: 4 Rex-3: [Abor 5-10 mr-5], [Brandon-8, Lily-7], [Joos-9, Alice-10]] Party: Reform, Candidates: Bobs of Ports: 5 Party: Reform, Candidates: Bobs of Ports: 5 Party: Reform, Candidates: Bobs of Ports: 5 Party: Independent Candidate, Candidates: Bobs of Ports: 5 Party: Independent Candidate, Candidates: Bobs of Ports: 5 Port CPL, the number is the candidate's order on the original list. For OPL, the number is the candidate's number of individual votes. Party: Independent Candidate, Candidates: Bobs of Ports: 5 Party: Independent Candidate, Candidates: Bobs of Ports: 5 Party: Independent Candidate, Candidates:				Т	
The regular part in Test_WriteSummary_CPL.ja va The party ballot results are as follows: Party: Independent Candidate, Number of Votes: 1 Party: Independent Candidate, Number of Votes: 1 Party: Green, Number of Votes: 1 Party: Green, Number of Votes: 2 Party: Republican, Number of Votes: 5 Party: Republican, Number of Votes: 3 The voting type is CPL.		Candidate=0, Green=1, Republican=2, Democratic=3, Reform=1} partyNames = {"Democratic", "Republican", "Reform", "Green", "Independent Candidate"} results = [{Mary=2, Jack=1}, {Sunny=4, Rex=3}, {Bob=6, Tom=5}, {Brandon=8, Lily=7}, {Joe=9, Alice=10}]	follows: Party: Independent Candidate, Number of Votes: 0 Party: Green, Number of Votes: 1 Party: Republican, Number of Votes: 2 Party: Democratic, Number of Votes: 3 Party: Reform, Number of Votes: 1 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 2 Jack: 1 Party: Republican, Candidates: Mary: 2 Jack: 1 Party: Republican, Candidates: Sunny: 4 Rex: 3 Party: Reform, Candidates: Bob: 6 Tom: 5 Party: Green, Candidates: Brandon: 8 Lily: 7 Party: Independent Candidate, Candidates: Joe: 9 Alice: 10 For CPL, the number is the candidate's order on the original list. For OPL, the number is the candidate's number of individual	Party: Independent Candidate, Number of Votes: 0 Party: Green, Number of Votes: 1 Party: Republican, Number of Votes: 2 Party: Democratic, Number of Votes: 3 Party: Reform, Number of Votes: 1 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 2 Jack: 1 Party: Republican, Candidates: Sunny: 4 Rex: 3 Party: Reform, Candidates: Bob: 6 Tom: 5 Party: Green, Candidates: Brandon: 8 Lily: 7 Party: Independent Candidate, Candidates: Joe: 9 Alice: 10 For CPL, the number is the candidate's order on the original list. For OPL, the number is the candidate's	
Test_WriteSummary_CPL.ja va The voting type is CPL. The party ballot results are as follows: Party: Independent Candidate, Party: Independent Candidate, Party: Green, Number of Votes: 1 Party: Green, Number of Votes: 2 Party: Republican, Number of Votes: 6 Party: Democratic, Number of Votes: 3 The party ballot results are as follows: Party: Independent Candidate, Number of Votes: 1 Party: Green, Number of Votes: 2 Party: Republican, Number of Votes: 5 Party: Republican, Number of Votes: 5 Party: Reform, Number of Votes: 3 The voting type is CPL. Party: Independent Candidate, Number of Votes: 1 Party: Republican, Number of Votes: 2 Party: Democratic, Number of Votes: 3 The voting type is CPL. Party: Independent Candidate, Number of Votes: 1 Party: Green, Number of Votes: 2 Party: Republican, Number of Votes: 5 Party: Democratic, Candidates: Mary: 20 Jack: 10 Party: Republican, Candidates: Sunny: 40 Party: Democratic, Number of Votes: 3 The voting type is called in the party in the party: Independent Candidate, Number of Votes: 1 Party: Green, Number of Votes: 1			Election Summary Report	Election Summary Report	Write success
votingType = "CPL" follows: Party: Independent Candidate, Number of Votes: 1 partyBallots = {Independent Candidate, Number of Votes: 1 Party: Green, Number of Votes: 1 Party: Green, Number of Votes: 2 Party: Republican, Number of Votes: 5 Party: Republican, Number of Votes: 3 Party: Republican, Number of Votes: 5 Party: Reform, Number of Votes: 3 Party: Republican, Number of Votes: 5 Party: Reform, Number of Votes: 3 Party: Independent Candidate, Number of Votes: 1 Party: Green, Number of Votes: 6 Party: Democratic, Number of Votes: 3 Party: Republican, Number of Votes: 5 Party: Reform, Number of Votes: 3 Party: Republican, Number of Votes: 3 Party: Republican, Number of Votes: 3 Party: Republican, Number of Votes: 3 Party: Reform, Number of Votes: 3 Party: Republican, Number of Votes: 3 Party: Reform, Number of Votes: 3 Party: Republican, Number of Votes: 3 Par		Test_WriteSummary_CPL.ja	The voting type is CPL.	The voting type is CPL.	
Candidate=1, Green=2, Republican=6, Democratic=5, Reform=3} Party: Green, Number of Votes: Party: Democratic, Number of Votes: 5 Party: Republican, Number of Votes: 5 Party: Republican, Number of Votes: 5 Party: Reform, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 The candidate ballot results are as follows: Party: Democratic, Candidates: Mary: 20 Party: Republican, Number of Votes: 3 Party: Republican, Number of Votes: 3 Party: Republican, Number of Votes: 3			follows: Party: Independent Candidate,	Party: Independent Candidate, Number of Votes: 1	
Votes: 6 partyNames = Party: Democratic, Number of {"Democratic", "Reform", "Reform", "Green", "Independent "Candidate"} Votes: 6 Party: Democratic, Number of Votes: 5 Party: Democratic, Candidates: Mary: 20 Party: Reform, Number of Votes: 3 Party: Democratic, Candidates: Mary: 20 Party: Republican, Candidates: Sunny: 40 Rex: 30		Candidate=1, Green=2, Republican=6,	Party: Green, Number of Votes: 2	Party: Republican, Number of Votes: 6 Party: Democratic, Number of Votes: 5	
{"Democratic", Votes: 5 Party: Democratic, Candidates: Mary: 20 "Republican", "Reform", Party: Reform, Number of User: 3 Party: Republican, Candidates: Sunny: 40 Party: 40 Party: Republican, Candidates: Sunny: 40 Party: 40 Party			Votes: 6		
"Green", "Independent Votes: 3 Party: Republican, Candidates: Sunny: 40 Rex: 30		{"Democratic",	Votes: 5	Party: Democratic, Candidates: Mary: 20	
Test write summary results to Candidate"} Rex: 30		"Republican", "Reform",	Party: Reform, Number of		
	Test write summary results to file for CPL	Candidate"}			

Jack=10}, {Sunny=40, Rex=30}, {Bob=60, Tom=50}, {Brandon=80, Lily=70}, {Joe=90, Alice=100}]	Party: Democratic, Candidates: Mary: 20 Jack: 10 Party: Republican, Candidates: Sunny: 40 Rex: 30 Party: Reform, Candidates: Bob: 60 Tom: 50 Party: Green, Candidates: Brandon: 80 Lily: 70 Party: Independent Candidate, Candidates: Joe: 90 Alice: 100 For CPL, the number is the candidate's order on the original list. For OPL, the number is the	Party: Green, Candidates: Brandon: 80 Lily: 70 Party: Independent Candidate, Candidates: Joe: 90 Alice: 100 For CPL, the number is the candidate's order on the original list. For OPL, the number is the candidate's number of individual votes.	
	candidate's number of individual votes.		

Post condition(s) for Test:
The election summary report is written to the file.

Project Name: Project 1: Voting System	Team# 12
Test Stage: Unit X System	Test Date: Nov. 16 2019
Test Case ID#: 008 Test Description: Test the "writeAuditFile" method to see if the election result is saved in the audit file	Name(s) of Testers: Xiaohui Chao
Automotoda vog no V	Indicate where are you storing the tests (what file) and the name of the method/functions being used. ./Test/Test_WriteAudit.java
Automated: yes no X Results: Pass X Fail	

Preconditions for Test: The data is ready to be written into the audit file. If the input data is string, it can be written directly to the audit file. If the input data is integer, it has been transferred to string.

Step #	Test Step Description	Test Data	_	Actual Result	Notes
		The regular part in Test_WriteAudit.java	CPL	CPL	Write success
1		str = CPL path = AuditFile1.txt			
		The regular part in Test_WriteAudit.java			Write success
2	\mathcal{E}	str = null path = AuditFile2.txt			
		The regular part in Test_WriteAudit.java			Write success
	Test write empty string to audit file	str = '''' path = AuditFile3.txt			

Post condition(s) for Test:		
The string is written to the audit file.		

Project Name: Project 1: Voting System Team# 12

Test Stage: Unit X System __ Test Date: Nov. 17 2019

Test Case ID#: 009 Name(s) of Testers: Xiaohui Chao

Test Description:
Test the "parse_opl" method to see if the csv file can be correctly parsed

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

./Test/Test_parser_opl.java

Automated: yes___ no X

Results: Pass X Fail_____

Preconditions for Test: The CSV file is in the same folder and ready to be parsed.

Step #	Test Step Description	Test Data	_	Actual Result	Notes
			[3, 30, [D, R, I], {I=5, R=11, D=14}, [{Pike=4, Foster=10},		Parse success
	OPL when there is no tie				

_				
Ī		1,,,,,		
		,1,,,,		
		,1,,,,		
		,,,,1,		
		,,,1,,		
		,,,,1		
		,,,1,,		
		,,,,1		
		,,1,,,		
		,,,,1,		
		,1,,,,		
		,,,1,,		
		1,,,,,		
		,,,,1		
		,1,,,,		
		,1,,,,		
		,,1,,,		
		,1,,,,		
		,1,,,,		
		,,,1,,		
		,,,,1		
		,1,,,,		
		,,,1,,		
		,1,,,,		
1				

		,,,,1,			
		,,,1,,			
		,,,,1			
		1,,,,,			
		,1,,,,			
		auditfile = "audit.txt"			
				[3, 30, [D, R, I], {I=5, R=11, D=14}, [{Pike=5, Foster=9}, {Deutsch=2, Walters=3,	Parse success
		OPL	{Deutsch=2, Walters=3, Borg=6}, {Smith=5}]]	Borg=6}, {Smith=5}]]	
			(**************************************		
		3			
		30			
		6			
		[Pike,D]			
		[Foster,D]			
		[Deutsch,R]			
		[Borg,R]			
		[Walters,R]			
		[Smith,I]			
		1,,,,,			
		1,,,,,			
		,1,,,,			
		,1,,,,			
		,,,,1,			
2	Test the parser function for OPL when there is a tie	,,,1,,			

	1		
	,,,,1		
	,,,1,,		
	,,,,1		
	,,1,,,		
	,,,,1,		
	,1,,,,		
	,,,1,,		
	1,,,,,		
	,,,,1		
	,1,,,,		
	,1,,,,		
	,,1,,,		
	,1,,,,		
	1,,,,,		
	,,,1,,		
	,,,,1		
	,1,,,,		
	,,,1,,		
	,1,,,,		
	,,,,1,		
	,,,1,,		
	,,,,1		
	1,,,,,		
	,1,,,,		

	auditfile = "audit.txt"			
	The data in test_opl_sameName.csv:	D=14}, [{Pike=4, Foster=10}, {Deutsch=2, Walters=3, Borg=6},	[3, 30, [D, R, I], {I=5, R=11, D=14}, [{Pike=4, Foster=10}, {Deutsch=2, Walters=3, Borg=6}, {Walters=5}]]	Parse success
	OPL 3 30 6 [Pike,D] [Foster,D] [Deutsch,R] [Borg,R] [Walters,R] [Walters,I] 1,,,,, 1,,,,,	{Walters=5}]]		
	,,,,1, ,,,,1, ,,,,,1, ,,,,,1, ,,,,,,,,,			
	,1,,,, ,,1,,, ,,1,,, ,,1,,, ,,1,,, ,,1,,, ,,1,,, ,,1,,, ,,1,,, ,,1,, ,,1,, ,,1,, ,,1,,			
Test the parser function for OPL when there is no tie but two candidates have the same name	,,,,,1 1,,,,, ,1,,,, auditfile = "audit.txt"			

Post condition(s) for Test:		
The csv file is correctly parsed		

Project Name: Project 1: Voting System	Team# 12		
Test Stage: Unit X System	Test Date: Nov. 18 2019		
Test Case ID#: 010 Test Description: Test the "parse_cpl" method to see if the csv file can be correctly parsed	Name(s) of Testers: Xiaohui Chao		
	Indicate where are you storing the tests (what file) and the name of the method/functions being used. ./Test/Test_parser_cpl.java		
Automated: yes no X			
Results: Pass X Fail			

Preconditions for Test: The CSV file is in the same folder and ready to be parsed.

Step #	Test Step Description	Test Data	•	Actual Result	Notes
		The data in test_cpl_reg.csv:	3 30 D R	3 30 D R	Parse success

	CPL 3 [D,R,I] 3 30 6 [Pike,D,1] [Foster,D,2] [Deutsch,R,1]	I {I=10, R=14, D=6} [{Pike=1, Foster=2}, {Deutsch=1, Walters=3, Borg=2}, {Smith=1}]	
	[Borg,R,2] [Walters,R,3] [Smith,I,1] 1,, 1,, ,1,		
	,,1 1,, ,,1 ,,1 ,,1 ,,1 ,,1		

		ī			
		,1,			
		1			
		1,,			
		,,1			
		,1,			
		,1,			
		,,1			
		,1,			
		,1,			
		,1,			
		,,1			
		,1,			
		,,1			
		,1,			
		,,1			
		,,1			
		,1,			
		1,,			
		,1,			
		auditfile = "audit.txt"			
			3		Parse success
		The data in test_cpl_tie.csv:	30 D	30 D	
		CPL	R	R	
		3 [D,R,I]	I {I=12, R=9, D=9}	I {I=12, R=9, D=9}	
		3	[{Pike=1, Foster=2}, {Deutsch=1,	[{Pike=1, Foster=2}, {Deutsch=1, Walters=3,	
		30	Walters=3, Borg=2}, {Smith=1}]	Borg=2}, {Smith=1}]	
		6 [Pike,D,1]			
	Test the parser function for CPL when there is a tie	[Foster,D,2]			
2	CPL when there is a tie	[Deutsch,R,1]			

Boog.R.2						
Walters.R.3 Smith.l.1			[Borg,R,2]			
Smith.l.			[Walters,R,3]			
1, 1, 1, 1, 1, 1, 1, 1,			[Smith I 1]			
auditfile = "audit.txt" 3			[5111111,1,1]			
auditfile = "audit.txt" 3			1,,			
auditfile = "audit.txt" 3			1,,			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			,,,1			
auditfile = "audit.txt" 3			1,,			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			1,,			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			"1			
auditfile = "audit.txt" 3			,, <u>1</u>			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			,,1			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			1,,			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			l.1.			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			,, <u>1</u>			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			1.,			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			,÷, 1			
auditfile = "audit.txt" 3			,,1			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			1			
auditfile = "audit.txt" 3			,1,			
auditfile = "audit.txt" 3			,,1			
auditfile = "audit.txt" 3			1.,			
auditfile = "audit.txt" 3			l ï			
auditfile = "audit.txt" 3			,,, <u>-</u>			
auditfile = "audit.txt" 3			1,,			
The data in test_cpl_sameName.csv: D R CPL I {I=10, R=14, D=6} {I[Pike=1, Foster=2}, {Deutsch=1, Walters=3}, Borg=2}, {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same R R R R R R R R R			,,1			
The data in test_cpl_sameName.csv: D R CPL I {I=10, R=14, D=6} {I[Pike=1, Foster=2}, {Deutsch=1, Walters=3}, Borg=2}, {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same R R R R R R R R R						
The data in test_cpl_sameName.csv: D R CPL I {I=10, R=14, D=6} {I[Pike=1, Foster=2}, {Deutsch=1, Walters=3}, Borg=2}, {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same R R R R R R R R R			auditfile = "audit tyt"			
The data in test_cpl_sameName.csv: D R CPL I [I=10, R=14, D=6] { [Pike=1, Foster=2], {Deutsch=1, Walters=3, Walters=3} } [PD,R,I] { Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same of the content of the conten	<u> </u>		auditiiic – audititxt		_	_
test_cpl_sameName.csv: D				Γ	3	Parse success
test_cpl_sameName.csv: D			The data in	30	30	
CPL R I I I I I I I I I I I						
{I=10, R=14, D=6} {Pike=1, Foster=2}, {Deutsch=1, [Pike=1, Foster=2], {Walters=3, Borg=2}, {Walters=1}]} Test the parser function for CPL when there is no tie but two candidates have the same {I=10, R=14, D=6} {{Pike=1, Foster=2}, {Deutsch=1, Walters=3, Borg=2}, {Walters=1}]} 3 3 4 Test the parser function for CPL when there is no tie but two candidates have the same 6			test_cpi_sameName.csv:	P	ש	
{I=10, R=14, D=6} {Pike=1, Foster=2}, {Deutsch=1, [Pike=1, Foster=2], {Walters=3, Borg=2}, {Walters=1}]} Test the parser function for CPL when there is no tie but two candidates have the same {I=10, R=14, D=6} {{Pike=1, Foster=2}, {Deutsch=1, Walters=3, Borg=2}, {Walters=1}]} 3 3 4 Test the parser function for CPL when there is no tie but two candidates have the same 6				R	R	
{I=10, R=14, D=6} {Pike=1, Foster=2}, {Deutsch=1, [Pike=1, Foster=2], {Walters=3, Borg=2}, {Walters=1}]} Test the parser function for CPL when there is no tie but two candidates have the same {I=10, R=14, D=6} {{Pike=1, Foster=2}, {Deutsch=1, Walters=3, Borg=2}, {Walters=1}]} 3 3 4 Test the parser function for CPL when there is no tie but two candidates have the same 6			CPL	lt	т	
3 [{Pike=1, Foster=2}, {Deutsch=1, Walters=3, Borg=2}, {Walters=1}] [D,R,I] {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same of the candidate of the candidat				(T. 10 D. 14 D. 1)	r 10 P 11 P 10	
Walters=3, Borg=2}, {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same						
Walters=3, Borg=2}, {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same			3	[{Pike=1, Foster=2}, {Deutsch=1	[{Pike=1, Foster=2}, {Deutsch=1, Walters=3	
[D,R,I] {Walters=1}] Test the parser function for CPL when there is no tie but two candidates have the same 6				Wolters 2 Dans 2)	Dorg-2) (Welters-1)]	
Test the parser function for CPL when there is no tie but two candidates have the same 6				wanters=5, borg=2},	Dorg=2}, { watters=1 }]	
Test the parser function for CPL when there is no tie but two candidates have the same 6			[D,R,I]	{Walters=1}]		
CPL when there is no tie but two candidates have the same 6						
CPL when there is no tie but two candidates have the same 6			2			
CPL when there is no tie but two candidates have the same 6			3			
CPL when there is no tie but two candidates have the same 6						
CPL when there is no tie but two candidates have the same 6			30			
CPL when there is no tie but two candidates have the same 6		rest the parser runetion for	50			
two candidates have the same 6		CPL when there is no tie but				
		two candidates have the same	6			
5 name						
	1 3	name				

	[Pike,D,1]		
	[Foster,D,2]		
	[Deutsch,R,1]		
	[Borg,R,2]		
	[Walters,R,3]		
	[Walters,I,1]		
	1,,		
	1,,		
	,1,		
	,1,		
	,,1		
	1,,		
	,,1		
	1,,		
	,1,		
	,,1		
	,,1		
	,1,		
	,1,		
	1,,		
	,,1		
	,1,		
	,1,		
	,,1		

	,1,		
	,1,		
	,1,		
	,,1		
	,1,		
	,,1		
	,1,		
	,,1		
	,,1		
	,1,		
	1,,		
	,1,		
	auditfile = "audit.txt"		

Post condition(s) for	Test:
------------------	-------	-------

The csv file is correctly parsed

Test Stage: System Test Date: Nov. 16 2019

Test Case ID#: 011 Name(s) of Testers: Rex Zhu

Test Description:

Test a regular CPL file

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_system

Automated: no

Results: Pass

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
	_				
	Turn on a terminal Change		Party: I, Number of Votes: 10	Election Summary Report	Successful! Audit file is good.
	directory to/src Run javac *.java		Party: R, Number of Votes: 14 Party: D, Number of Votes: 6	The voting type is CPL.	Audit files are checked! Here are the audit files:
	Run java -d/testing *.java	toot and man age			https://github.com/zzdpk2/CS
1				Party: I, Number of Votes: 10 Party: R, Number of Votes: 14	CI5801/tree/master/cplandresu
	C1 4- /44:			Party: D, Number of Votes: 6	
	Cd to /testing Run javac *.java				
2	T	test_cpl_reg.csv		The candidate ballot results are as follows: Party: D, Candidates: Pike: 1	
3	Type csv filename	test_cpl_reg.csv		Party: R, Candidates: Deutsch: 1	
				Party: I, Candidates: Smith: 1	
				For CPL, the number is the candidate's	
				order on the original list.	
	T (1 (1 1	4 4 1		For OPL, the number is the candidate's	
4	Type filename for saving result	test_cpl_reg.csv		number of individual votes.	

Post condition(s) for Test:		
The result is correct.		

Test Stage: System Test Date: Nov. 16 2019

Test Case ID#: 012 Name(s) of Testers: Rex Zhu

Test Description:

Test the CSV file that contains item with the same names

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_system

Automated: no

Results: Pass

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Turn on a terminal Change directory to/src Run javac *.java Run java -d/testing *.java		Party: I, Number of Votes: 10 Party: R, Number of Votes: 14 Party: D, Number of Votes: 6 Party: D, Candidates: Pike: 1 Party: R, Candidates: Deutsch: 1 Party: I, Candidates: Walters: 1	Election Summary Report The voting type is CPL. The party ballot results are as follows: Party: I, Number of Votes: 10 Party: R, Number of Votes: 14	Successful! Audit file is good. Audit files are checked! Here are the audit files: https://github.com/zzdpk2/CS CI5801/tree/master/cplandresu lt
2	Cd to /testing Run javac *.java Run java Test_system Type csv filename			Party: D, Number of Votes: 6 The candidate ballot results are as follows: Party: D, Candidates: Pike: 1 Party: R, Candidates: Deutsch: 1	
4	Type filename for saving result	test_cpl_sameName .csv		Party: I, Candidates: Walters: 1 For CPL, the number is the candidate's order on the original list. For OPL, the number is the candidate's number of individual votes.	

Post condition(s) for Test:
The result is correct.

Test Stage: System Test Date: Nov. 16 2019

Test Case ID#: 013 Name(s) of Testers: Rex Zhu

Test Description:

Test the CSV that has ties

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_system

Automated: no

Results: Pass

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
				Election Summary Report	Successful!
	Turn on a terminal Change		Party: R, Number of Votes: 9		Audit file is good.
	directory to/src Run javac *.java		Party: D, Number of Votes: 9	The voting type is CPL.	Audit files are checked! Here are the audit files:
	Run java -d/testing *.java		Party: D, Candidates: Pike: 1	The party ballot results are as follows:	https://github.com/zzdpk2/CS
1				Party: I, Number of Votes: 12	CI5801/tree/master/cplandresu
1				Party: R, Number of Votes: 9	lt
	Cd to /testing			Party: D, Number of Votes: 9	
	Run javac *.java			The candidate ballot results are as follows:	
2	Run java Test_system			Party: D, Candidates: Pike: 1	
3	Type csv filename			Party: R, Candidates: Deutsch: 1	
	57			Party: I, Candidates: Smith: 1	
				and it is a contraction of the c	
				For CPL, the number is the candidate's	
				order on the original list.	
		test_cpl_tie.csv		For OPL, the number is the candidate's	
4	Type filename for saving result			number of individual votes.	

Post condition(s) for Test:
The result is correct.

Team# 12

Test Stage: System Test Date: Nov. 16 2019

Test Case ID#: 014 Name(s) of Testers: Rex Zhu

Test Description:

Test OPL regular CSV file.

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_system

Automated: no

Results: Pass

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Turn on a terminal Change directory to/src Run javac *.java Run java -d/testing *.java Cd to /testing Run javac *.java		Party: R, Number of Votes: 11 Party: D, Number of Votes: 14 Party: D, Candidates: Foster: 10	Election Summary Report The voting type is OPL. The party ballot results are as follows: Party: I, Number of Votes: 5 Party: R, Number of Votes: 11 Party: D, Number of Votes: 14 The candidate ballot results are as follows:	Successful! Audit file is good. Audit files are checked! Here are the audit files: https://github.com/zzdpk2/CS C15801/tree/master/cplandresu lt
	Run java Test_system Type csv filename			Party: D, Candidates: Foster: 10 Party: R, Candidates: Borg: 6	
4	Type filename for saving result	test_opl_reg.csv		Party: I, Candidates: Smith: 5 For CPL, the number is the candidate's order on the original list. For OPL, the number is the candidate's number of individual votes.	

Post condition(s) for Test:		
The result is correct.		

Test Stage: System Test Date: Nov. 16 2019

Test Case ID#: 015 Name(s) of Testers: Rex Zhu

Test Description:

The CSV file in OPL that has items with the same name

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_system

Automated: no

Results: Pass

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
			Party: I, Number of Votes: 5	Election Summary Report	Successful!
	Turn on a terminal Change		Party: R, Number of Votes: 11		Audit file is good.
	directory to/src		Party: D, Number of Votes: 14	The voting type is OPL.	Audit files are checked!
	Run javac *.java			TT	Here are the audit files:
	Run java -d/testing *.java			1 1 2	https://github.com/zzdpk2/CS
1			Party: R, Candidates: Borg: 6 Party: I, Candidates: Walters: 5	Party: I, Number of Votes: 5 Party: R, Number of Votes: 11	CI5801/tree/master/cplandresu
			Farty. 1, Candidates. Waiters. 5	Party: D, Number of Votes: 14	II.
	Cd to /testing			arty. D, rumber of votes. 14	
	Run javac *.java			The candidate ballot results are as follows:	
2	Run java Test_system			Party: D, Candidates: Foster: 10	
3	Type csv filename			Party: R, Candidates: Borg: 6	
				Party: I, Candidates: Walters: 5	
				E CDI di li di li di li	
				For CPL, the number is the candidate's	
		test_opl_sameName		order on the original list. For OPL, the number is the candidate's	
4	Type filename for saving result	•		number of individual votes.	

Post condition(s) for Test:		
The result is correct.		

Test Stage: System Test Date: Nov. 16 2019

Test Case ID#: 016 Name(s) of Testers: Rex Zhu

Test Description:

Test the CSV in OPL with tie

Indicate where are you storing the tests (what file) and the

name of the method/functions being used.

./Test/Test_

Automated: no

Results: Pass _____

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
			Party: I, Number of Votes: 5	Election Summary Report	Successful!
	Turn on a terminal Change		Party: R, Number of Votes: 11		Audit file is good.
	directory to/src		Party: D, Number of Votes: 14	The voting type is OPL.	Audit files are checked!
	Run javac *.java				Here are the audit files:
	Run java -d/testing *.java		Party: D, Candidates: Foster: 9	The party ballot results are as follows:	https://github.com/zzdpk2/CS
1			Party: R, Candidates: Borg: 6	Party: I, Number of Votes: 5	CI5801/tree/master/cplandresu
1			Party: I, Candidates: Smith: 5	Party: R, Number of Votes: 11	lt
	Cd to /testing			Party: D, Number of Votes: 14	
	Run javac *.java			The 1: 1-4- b-11-414 f-11	
2	Run java Test_system			The candidate ballot results are as follows:	
3	Type csv filename			Party: D, Candidates: Foster: 9 Party: R, Candidates: Borg: 6	
	Type est menane			Party: I, Candidates: Smith: 5	
				arty. 1, Candidates. Simui. 3	
				For CPL, the number is the candidate's	
				order on the original list.	
		test_opl_tie.csv		For OPL, the number is the candidate's	
4	Type filename for saving result			number of individual votes.	

Post condition(s) for Test:		
The result is correct.		