Voting System (BallotTest) Team# 7

Test Stage: UNIT	**Test Date:** 3/27/23
Test Case ID#: BallotTest_1	**Name(s) of Testers:** Lucas Olsen (olse0280)
Test Name: addChoice	**Description:** Add a choice to the ballot and have it returned with getChoice(). Call getChoice() again to make sure only 1 choice was added.
Automated: YES	**Test location:** Executable from ‘make BallotTest` or `make tests` compiles to /src/gtest_code/executables/BallotTest
Results: PASS	
Preconditions: Compile executable with `make BallotTest` or `make test` from the /project1/ directory Makefile	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	b.addChoice(1) b.getChoice()	Ballot b (1)	1	1	getChoice removes the returned choice. b is now empty
2	b.getChoice()	Ballot b (empty)	-1	-1	getChoice returns -1 on error (when the ballot is empty)

Post condition(s) for Test:

Ballot b will be empty

Voting System (BallotTest) Team# 7

Test Stage: UNIT	**Test Date:** **3/27/23**
Test Case ID#: BallotTest_2	**Name(s) of Testers:** Lucas Olsen (olse0280)
Test Name: EmptyChoices	**Description:** Call getChoice on an empty ballot
Automated: YES	**Test location:** Executable from ‘make BallotTest` or `make tests` compiles to /src/gtest_code/executables/BallotTest
Results: **PASS**	
Preconditions: Compile executable with `make BallotTest` or `make test` from the /project1/ directory Makefile	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	b.getChoice()	Ballot b (empty)	-1	-1	getChoice returns -1 on error (when the ballot is empty)

Post condition(s) for Test:

Ballot b will be empty

Voting System (BallotTest) Team# 7

Test Stage: UNIT	**Test Date:** **3/27/23**
Test Case ID#: BallotTest_3	**Name(s) of Testers:** Lucas Olsen (olse0280)
Test Name: addChoiceOrder	**Description:** add 10 random choices to a ballot using addChoice(). Test their expected values against an array of the same random data using getChoice()
Automated: YES	**Test location:** Executable from ‘make BallotTest` or `make tests` compiles to /src/gtest_code/executables/BallotTest
Results: **PASS**	
Preconditions: Compile executable with `make BallotTest` or `make test` from the /project1/ directory Makefile	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Random number generation	Ballot b int arr[10]	Ballot b (a, b, … j), arr{a, b, …. j} *a-j are random numbers stored in the same order*	–	Unable to view private variables. arr should contain the same variables as b in the same ordering. This is tested in the next step
2	*for i=0 to i=10* b.getChoice()	Ballot b int arr[10]	b.getChoice() == arr[0] b.getChoice() == arr[1] … b.getChoice() == arr[9]	b.getChoice = arr[0] b.getChoice = arr[1] … b.getChoice() = arr[9]	

Post condition(s) for Test:

Ballot b will be empty