|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team#** | |
| **Test Stage: Unit √\_\_ System \_\_** | **Test Date: Nov. 17th 2019** |
| **Test Case ID#: 002** | **Name(s) of Testers: Zhuoran Bi** |
| **Test Description:**  Test the functionality of the increment() in CPL class, to see if this function can increment votes for a party for a given party index. |  |
| **Automated: yes\_√\_ no \_\_\_** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.** |
| **Results: Pass \_\_\_\_\_ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:**  The processer has already separated the given input file to a CPL file, and result has been generated. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  |  |  |  |  |  |
| 1 | To test increment() function, for each party, we increment the votes for that party once and check the number of votes for that party and other parties | Input data1 |  |  | Test correct |
| 2 | To test increment() function can handle edge cases. For example, negative index of parties and the index exceeding the actual number of parties | Input data1 | -1 |  | Test correct |
| 3 | To test increment() function, for each party, we increment the votes for that party once and check the number of votes for that party and other parties | Input data2 | 1,0,0,2,0,2,0,1,0 |  | Test correct |
| 4 | To test increment() function can handle edge cases. For example, negative index of parties and the index exceeding the actual number of parties | Input data2 | -1 |  | Test correct |

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

The test script should output some error messages if there are errors occur. The state for a specific party should change since the votes for that specific party have incremented while others should not change.

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 notesfor you and your team members.