|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: Main\_1** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test for main test by import .csv file and check the data from the .csv whether or not correct. Main\_test.java store in /src/test folder. The method called test\_for\_IR\_file\_setting().** |  |
| **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: Import IR\_tie\_for\_winner.csv file and check the data structures.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test election type | Election type | IR | IR |  |
| 3 | Test number of candidates | Number of candidates | 4 | 4 |  |
| 4 | Test candidate’s’ name list | Array of candidate names | “rose”, “Kleinberg”, “Chou” , Royce” | “rose”, “Kleinberg”, “Chou” , Royce” | Using toarray() |
|  | Test number of ballot votes | Ballot votes | 7 | 6 |  |
|  | Test ballot array | String Array of ballot | “1,2,4,2” “1,,2,” “1,2,3,” “3,2,1,4” “,,1,2” “,,,1” “2,,1,” | “1,2,4,2” “1,,2,” “1,2,3,” “3,2,1,4” “,,1,2” “,,,1” “2,,1,” | Using toarray() |

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

No if-else condition. Just check reading file and generate data structure.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: Main\_2** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test for main test by import .csv file and check the data from the .csv whether or not correct. Main\_test.java store in /src/test folder. The method called test\_for\_OPL\_file\_setting().** |  |
| **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: Import 6\_OPL.csv file and check the data structures.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Test ballot array | String Array of ballot | “1,,,,,” “1,,,,,” “,1,,,,” “,,,,1” “,,,1,,” “,,,1,,” “1,,,,,” “,1,,,,” | “1,,,,,” “1,,,,,” “,1,,,,” “,,,,1” “,,,1,,” “,,,1,,” “1,,,,,” “,1,,,,” |  |
| 2 | Test election type | Election type | “OPL” | “OPL” |  |
| 3 | Test number of candidates | Number of candidates | 6 | 6 |  |
| 4 | Test array of number candidates and separate their party | Test array candidates name and their parties | “Pike” “D” “Foster” “D” “Deutsch” “R” “Borg” “R” “Jones” “R” “Smith” “D” | “Pike” “D” “Foster” “D” “Deutsch” “R” “Borg” “R” “Jones” “R” “Smith” “D” |  |
|  | Test number of parties | Number of parties | 3 | 3 |  |
|  | Test number of ballot votes | Ballot votes | 9 | 9 |  |

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

No if-else branch in this test. Just check reading file and generate data structure.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: Main\_3** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Call execute\_IR to test result of election.**  **Execute\_IR() in main\_test** |  |
| **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: Create the ballot and number of ballot and candidates list as the input of method.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Test winner of election | election winner | “Rose” | “Rose” |  |
| 2 | Test getcaancidate getter | length of getcandidate | 4 | 4 |  |
| 3 | Test getBallot getter | length Getballot | 4 | 4 |  |
| 4 | Test number of ballot getter | Getnum\_ballot | 4 | 4 |  |
|  | Test number of candidate getter | Get\_candidate | 4 | 4 |  |
|  | Test election type | Election type | “IR” | “IR” |  |

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

There is no if-else statement, so all branch converage.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: Main\_4** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Call execute\_OPL to test result of election.**  **Execute\_OPL() in main\_test** |  |
| **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:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Get length of ballot list | length of ballot | 9 | 9 |  |
| 2 | Test election type | Election type | “OPL” | “OPL” |  |
| 3 | Test number of candidate getter | Get\_candidate | 6 | 6 |  |
| 4 | Test length of array contains number of candidate and their parties | Length of get\_candidate | 12 | 12 |  |
|  | Test getBallot getter | Test getBallot getter | 9 | 9 |  |
|  | Test number of seats | Number of seats | 3 | 3 |  |

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

There is no if-else statement, so all branch converage.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_1** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: test the data store in the map.**  test\_store\_ballot\_data() |  |
| **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: call the ballot list and candidate and empty map.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Treverse every candicate’s ballot and check the data store | List of Rose’s ballot store | “10,30,40,20” “10,0,20,0” “10,20,30,0” | “10,30,40,20” “10,0,20,0” “10,20,30,0” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

Just make sure the ballot have stored in the arraylist. Give the example of Rose, cause its very complex. And all the information has stored.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_2** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test winner method if its succeed.**  test\_winner\_True() |  |
| **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: create ballots and candidate and maps that put key and value in map as parameter to call the function** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test whether or not successful winner comes out. | Boolean type of winner() | True | True |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

There is no if-else statement, all the branches have tested.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_3** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test winner method if its failed.**  test\_winner\_False() |  |
| **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: create ballots and candidate and maps that put key and value in map as parameter to call the function** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test whether or not successful winner comes out. | Boolean type of winner() | False | False |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

There is no if-else statement, all the branches have tested.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_4** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: test of name of winner.**  test\_name\_of\_winner() |  |
| **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: create the hash map and put value and key into the map.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test map of size | Map size | 4 | 4 |  |
| 3 | Test winner’s name | String type of winner name | “Rose” | “Rose” |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

My winner of name by this case is “Rose” all the branches covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_5** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test when the candidate loses, transfer his ballot to other candidates. Check the ballot**  test\_transfer() |  |
| **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: Create hash map and candidate list, and giving the lose candidate name checking one complex example of ballot transfer.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test elements array of ballot | Test elements array of transfer ballot | “0,10,0,0” “0,20,0,10” | “0,10,0,0” “0,20,0,10” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

Ballot has been transferred and get the correct array of current ballot. All the branch covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_6** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test when the candidate name loser. For this case only one candidate lose**  test\_loser\_name\_1() |  |
| **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: Create map and put keys and values into the map, create file as parameters to check the loser name whether or not correct.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test name of loser name | Test String of loser name | Cat | Cat |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

Lost name has successfully return by the method, and all branches have covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_7** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test when the candidate name loser. For this case has two candidate lose and random (filp coin) to choose loser**  test\_loser\_name\_2() |  |
| **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: Create map and put keys and values into the map, create file as parameters to check the loser name whether or not correct.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test name of loser name | Test name of loser name | “Cat” or “Tiger” | “Cat” or “Tiger” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

Lost name has successfully return by the method, and all branches have covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: IR\_8** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test for flip coin when two candidates in the final round and only one candidate win.**  test\_flip\_coin() |  |
| **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: Create hash map and put keys and values into the map.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test name of winner by flip coin | name of winner by flip coin | “Rose” or “Kleinberg” | “Rose” or “Kleinberg” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

Winner name has successfully return by the method, and all branches have covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: GF\_1** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: test whether or not generated/created file**  test\_create() |  |
| **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: Create file name and check whether or not is file** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test file type | File type | true | true |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered, the file when I call the createfile() method, its generate file successfully.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: GF\_2** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: Test the file writer when you write into the file test the output by reading.**  test\_file\_writer() |  |
| **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: Generate new file and write something call write() “Hello world”** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test writing item | String of writring item | “Hello world” | “Hello world” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered, the file when I call the file\_writter() method, its writing item is “Hello World”

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_1** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: test election type for OPL**  test\_OPL\_election() |  |
| **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: Call the OPL\_election() create names of candidates and their parties, and their ballots and number of seat and total number of ballot call this function.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test return value to show the election completed | String of return value | “OPL Done!” | “OPL Done!” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The result is correct and all branches covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_2** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Test parties and number of candidate**  test\_party\_candidates() |  |
| **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: create the map and put key and value into the map, and generate file as arguments can call the method()** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test number candidates in the parties | Map are not same | map | map |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test: After running the method, the maps are same. All the branches covered.**

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_3** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Septate candidates and their parties and return the string array of their names.**  test\_candidates\_list |  |
| **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: creating their names and parties and call the method() to septate** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test element of candidate name array | element of candidate name array | “Pike Foster Deutsch Borg Jones smith” | “Pike Foster Deutsch Borg Jones smith” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The output array is correct and all the branches have covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_4** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description:** test candidates number of vote  test\_candidates\_numOfVote |  |
| **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: c**reate map and put keys and values into the map. Then call the method | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Compared maps whether or not match | Compared maps whether or not match | map | map |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The output map is correct and all the branches have covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_5** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Test party votes**  test\_party\_vote() |  |
| **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: Create maps and put keys and values into the map and each candidate vote into the another map** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Check return map whether or not correct | map | map | map |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The return map is exactly same.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_6** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: T**est first allocation  test\_first\_allocation() |  |
| **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: c**reate map and put key and value into the map and call the method. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Check return map whether or not correct | map | map | map |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The return map is exactly same.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_7** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: test reminder.**  test\_reminder |  |
| **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: Create map and put keys and value into the map and call the function.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test reminder map | reminder map | map | map |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The return map is exactly same.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_8** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Test second allocation**  second\_allocation |  |
| **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: c**reate map and put key and value into the map and call the method. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test map whether or not contains key | Contains keys | true | true |  |
| 3 | Check value of key | value of key intertion | map | map |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The return map is exactly same. Contain all the key, and all the branches covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_9** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Test f**inal **allocation**  test\_final\_allocation() |  |
| **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: c**reate map and put key and value into the map and call the method. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test map whether or not contains key | Contains keys | true | true |  |
| 3 | Check value of key | value of key intertion | map | map |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

The return map is exactly same. Contain all the key, and all the branches covered.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_10** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Test flip coin method to determine who win the election when two candidate tie.**  test\_flip\_coin\_winner() |  |
| **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:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test name of candidate | name of candidate | “Tom” or “Jerry” | “Tom” or “Jerry” |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit ✅ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: OPL\_11** | **Name(s) of Testers: Yiming Yao/Hao wu** |
| **Test Description: Test printout candidate winner**  test\_candidates\_winner() |  |
| **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: Set up before test, and create map and put keys and values into the map and call the function** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test system out string | System out string | For Party D, Foster Pike won the election | For Party D, Foster Pike won the election |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

System out string is correct that means method output correct.

Winner name has successfully return by the method, and all branches have covered

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit System ✅** | **Test Date: 3/13/2021** |
| **Test Case ID#: System\_1** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: This is test for entire system, just put the .csv file and automatedly comes out result by terminal. This is for the More\_data\_IR.csv**  test\_program1() |  |
| **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: copy the testing file to right directory (from readme file)** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test result from output | Result from output | ---> winner is H\n" +  "---> For more details, please check result and audit file under /IR.\n" +  "---> Have a good day! | ---> winner is H\n" +  "---> For more details, please check result and audit file under /IR.\n" +  "---> Have a good day! |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered. The test passes.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit System ✅** | **Test Date: 3/13/2021** |
| **Test Case ID#: System\_2** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: This is test for entire system, just put the .csv file and automatedly comes out result by terminal. This is for the 10000data\_IR.csv**  test\_program\_2() |  |
| **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: copy the testing file to right directory (from readme file)** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test result from output | Result from output | ---> winner is F\n" +  "---> For more details, please check result and audit file under /IR.\n" +  "---> Have a good day! | ---> winner is F\n" +  "---> For more details, please check result and audit file under /IR.\n" +  "---> Have a good day! |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered. The test passes.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit System ✅** | **Test Date: 3/13/2021** |
| **Test Case ID#: System\_3** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: This is test for entire system, just put the .csv file and automatedly comes out result by terminal. This is for the 1000data\_IR.csv** |  |
| **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: copy the testing file to right directory (from readme file)** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test result from output | Result from output | ---> winner is D\n" +  "---> For more details, please check result and audit file under /IR.\n" +  "---> Have a good day! | ---> winner is D\n" +  "---> For more details, please check result and audit file under /IR.\n" +  "---> Have a good day! |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered. The test passes.

All the branches covered. The test passes.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit System ✅** | **Test Date: 3/13/2021** |
| **Test Case ID#: System\_4** | **Name(s) of Testers: Yiming Yao** |
| **Test Description:**  **This is test for entire system, just put the .csv file and automatedly comes out result by terminal. This is for the 6\_IR.csv** |  |
| **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: copy the testing file to right directory (from readme file)** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 |  |  |  |  |  |
| 2 | Test result from output | Result from output | "For Party R, Borg won the election\n" +  "For Party D, Foster Pike won the election\n" +  "For Party I, won the election\n" +  "---> For more details, please check result and audit file under /OPL.\n" +  "---> Have a good day!" | "For Party R, Borg won the election\n" +  "For Party D, Foster Pike won the election\n" +  "For Party I, won the election\n" +  "---> For more details, please check result and audit file under /OPL.\n" +  "---> Have a good day!" |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered. The test passes.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 22** | |
| **Test Stage: Unit System ✅** | **Test Date: 3/13/2021** |
| **Test Case ID#: System\_5** | **Name(s) of Testers: Yiming Yao** |
| **Test Description: This is test for entire system, just put the .csv file and automatedly comes out result by terminal. This is for the 11\_OPL.csv** |  |
| **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: copy the testing file to right directory (from readme file)** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Test result from output | Result from output | "For Party R, Deutsch won the election\n" +  "For Party D, Foster Pike won the election\n" +  "For Party I, won the election\n" +  "---> For more details, please check result and audit file under /OPL.\n" +  "---> Have a good day!" | "For Party R, Deutsch won the election\n" +  "For Party D, Foster Pike won the election\n" +  "For Party I, won the election\n" +  "---> For more details, please check result and audit file under /OPL.\n" +  "---> Have a good day!" |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

All the branches covered. The test passes. All good!