|  |  |
| --- | --- |
| **Project Name: Project 1: CompuVote Team #19** | |
| **Test Stage:** Unit | **Test Date:** 3/27/21 |
| **Test Case ID #:** test\_395\_07\_getLowestHighestCandidates\_03\_testGetLowestHighestCandidatesMultipleHighest | **Name(s) of Testers:** Aaron Kandikatla, Jack Fornaro |
| **Test Description:**  Given a standard IR election with candidates, parties, and ballots, where there is one clear candidate with the lowest votes and many candidates with the highest votes, tests that getLowestHighestCandidates returns only 1 candidate with highest votes and one highest candidate, despite there being many. |  |
| **Automated:** Yes | **Indicate where you are storing the tests (what file) and the name of the method/functions being used:**  Test file: Project1/src/test/org/team19/InstantRunoffSystemTest.java  Test method: testGetLowestHighestCandidatesMultipleHighest  Method/constructor being tested: getLowestHighestCandidates from Project1/src/main/org/team19/InstantRunoffSystem.java |
| **Results:** Pass |  |
|  |  |
| **Preconditions for Test:**  N/A | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  |  |  |  |  |  |
| 1 | Tests that only Rosen (highest candidate who was entered into the system first) is returned for highest candidate pair | Candidates:  Rosen (D) – 3 ballots  Kleinberg (R) – 0 ballots  Chou (I) – 2 ballots  Royce (L) – 1 ballot  Bobster (I) – 3 ballots | "Pair{Pair{0, [Kleinberg (R)]}, Pair{3, Rosen (D)}}" | "Pair{Pair{0, [Kleinberg (R)]}, Pair{3, Rosen (D)}}" | N/A |

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

N/A