

Test Cases for Coordinate Parsing in Sudoku Program

Test Case 1: End Case (A1)

- **Input:** "A1"
 - **Expected Output:**
 - $x = 0$
 - $y = 0$
 - **Test Type:** Boundary Condition
 - **Explanation:** This checks the parsing of the first valid coordinate at the top-left corner (min x and min y).
-

Test Case 2: Lowercase Input (a1)

- **Input:** "a1"
 - **Expected Output:**
 - $x = 0$
 - $y = 0$
 - **Test Type:** Requirement
 - **Explanation:** Tests the system's handling of lowercase input, which should still be valid as equivalent to uppercase.
-

Test Case 3: Backwards Input (1a)

- **Input:** "1a"
 - **Expected Output:**
 - $x = 0$
 - $y = 0$
 - **Test Type:** Requirement
 - **Explanation:** Tests a backwards coordinate format (number first, letter second) which should still be considered valid.
-

Test Case 4: Equivalence Test for A1, a1, and 1a

- **Input:** "A1", "a1", "1a"
- **Expected Output:**

- All inputs should result in $x = 0$ and $y = 0$
 - **Test Type:** Requirement
 - **Explanation:** Verifies that all three representations of the same coordinate result in identical parsing.
-

Test Case 5: Middle Case (E5)

- **Input:** "E5"
 - **Expected Output:**
 - $x = 4$
 - $y = 4$
 - **Test Type:** Requirement
 - **Explanation:** Tests a coordinate near the middle of the Sudoku grid.
-

Test Case 6: End Case (I9)

- **Input:** "I9"
 - **Expected Output:**
 - $x = 8$
 - $y = 8$
 - **Test Type:** Boundary Condition
 - **Explanation:** This tests the parsing of the bottom-right corner of the grid (max x and max y).
-

Test Case 7: Valid Coordinate with Different Format (B6)

- **Input:** "B6"
 - **Expected Output:**
 - $x = 5$
 - $y = 1$
 - **Test Type:** Requirement
 - **Explanation:** Tests the validity of coordinates where the letter and number aren't "even" with each other (don't convert to the same number).
-

Test Case 8: Valid Coordinate List Validation

- **Input:** ["A1", "a1", "1a", "E5", "I9", "B6"]
 - **Expected Output:**
 - All coordinates should have `"parsed_valid" == True`
 - **Test Type:** Requirement
 - **Explanation:** Verifies that all previously tested coordinates are marked as valid.
-

Test Case 9: Invalid Input (hello world)

- **Input:** "hello world"
 - **Expected Output:**
 - `"parsed_valid" == False`
 - **Test Type:** Error State
 - **Explanation:** Tests the scenario where the input contains more than 2 characters, which is invalid.
-

Test Case 10: Invalid Coordinate (A0)

- **Input:** "A0"
 - **Expected Output:**
 - `"parsed_valid" == False`
 - **Test Type:** Error State
 - **Explanation:** This tests for the case where a coordinate contains an invalid number (0 is not allowed).
-

Test Case 11: Invalid Coordinate (J1)

- **Input:** "J1"
- **Expected Output:**
 - `"parsed_valid" == False`
- **Test Type:** Error State
- **Explanation:** Tests for an invalid coordinate where the letter exceeds the valid range (A-I).

NOTE: The above formatting and text was generated by OpenAI's ChatGPT when asked to format the given assert statements into a more human-readable format. The creative work is my own, and the above text was reviewed and edited after generation.