Test Plan for count\_dead\_ants function

Document 1: Test Plan

Objective: Verify the functionality of the count\_dead\_ants function and ensure it accurately estimates the number of dead ants based on different input strings.

Scope: This test plan covers all parts of the function, including counting intact ants, heads, bodies, and dead ants using both methods.

Success Criteria:

All positive test cases pass with the expected dead ant count.

All negative test cases fail gracefully or identify issues with the function's logic.

No unexpected exceptions or errors occur during testing.

Test Environment:

Python 3.x interpreter

Any text editor or IDE

Test Data:

A variety of input strings representing different ant carnage scenarios, including:

Strings with various numbers of intact ants, extra heads, and unmatched body parts.

Empty strings and strings with no ants.

Strings with invalid characters or unexpected formats.

Test Execution:

Write individual test cases for each scenario with expected results.

Use a testing framework (e.g., unittest) or manual execution to run the test cases.

Verify the actual results against the expected results.

Document any failures or discrepancies.

Reporting:

Create a test report summarizing the test cases, results, and any identified issues.

Document 2: Positive Test Scenarios

\*\*Test Case ID Description Input String Expected Dead Ants Notes\*\*

1 No dead ants "ant...ant..." 0 All ants intact.

2 Dead ants only from extra heads "a..antb..ant" 1 One extra head indicates one dead ant.

3 Dead ants only from unmatched body parts "...ant.b.b.ant..." 2 Two unmatched bodies belong to separate dead ants.

4 Mixed dead ants from heads and body parts "ant.abb.anta..t.b" 3 Two extra heads and one unmatched body indicate three dead ants.

5 Empty string "" 0 No ants present.

Document 3: Negative Test Scenarios

\*\*Test Case ID Description Input String Expected Behavior Notes\*\*

1 Invalid characters "123ant456" Raise exception or return appropriate error Function should not handle non-alphanumeric characters.

2 Unexpected format "ant ant" Raise exception or return appropriate error Function expects a single string, not multiple separated ants.

3 All parts unmatched "abbabba" Raise exception or return appropriate error Function cannot determine dead ants without "ant" substring.

4 More body parts than possible ants "ant...bbb" Raise exception or return appropriate error More body parts than possible ants indicate invalid input.

These are just some examples, and you can create additional test cases based on your specific needs and understanding of the function's logic. Remember to adapt the test data and expected results accordingly.

By creating these documents, you have a clear plan for testing the count\_dead\_ants function and ensuring its accuracy and robustness.