# BlackBox Test Debugging (MS04 Submission)

**Bugs Found**:

We have identified that the testing function Test\_printMap() is not functioning as intended. Even when valid positions are inputted, the test case fails to pass. Consequently, we need to conduct a thorough evaluation of this function to determine the root cause of the failure.

Upon investigation, it has become clear that there are several issues within the testing function that need to be addressed. As a response, we have extended the function to improve its reliability and accuracy.

It’s important to note that while the correct implementation for blackbox testing is part of MS04 and not MS03, our group has opted to continue the development process this week. We plan to push these changes to the repository later, once the whitebox testing for MS04 is finalized.

A screenshot of a computer program

Description automatically generated

Testing output for further development

# A screenshot of a computer program Description automatically generated

# Black Box Test Debugging Continue(MS04 Submission)

**Test Name or ID:** test\_PointStructure(), test\_FunctionParameterType(), test\_DistanceCalculation(), test\_RouteSizeLimit(), test\_ShortestPathDocumentation(), test\_ProcessInput()

**Test Type:** Black Box

As stated above, black box testing has been conducted on various aspects of the code related to point structures, function parameters, distance calculations, route size limits, documentation consistency, and error handling. The identified issues have been thoroughly evaluated, and all test cases have been documented for further review and correction.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Description** | **Test Data** | | **Expected Result** | | **Actual Result** | | **Pass/Fail** | |
| **Test Point Structure** |  |  |  | Expected: Compilation should succeed with correct types for row and col. | | Actual: Compilation failed due to incorrect types for row and col. | | FAIL | |
|  |
| **Test Function Parameter Type** |  |  |  | Expected: Function should accept a pointer to struct Point. | | Actual: Function failed to compile due to incorrect parameter type. | | FAIL | |
|  |
| **Test Distance Calculation** |  |  |  | Expected: Correct distance calculated without overflow. | | Actual: Incorrect distance calculated due to potential overflow. | | FAIL | |
|  |
| **Test Route Size Limit Handling** |  |  |  | Expected: Should prevent overflow when adding points. | | Actual: No overflow check implemented; overflow occurred. | | FAIL | |
| **Test Process Input Error Handling** |  |  |  | Expected: Should return specific error codes based on conditions. | | Actual: Implementation did not return expected error codes. | | FAIL | |
| **Test Shortest Path Documentation** |  |  |  | Expected: Signature should match expected behavior in implementation. | | Actual: Signature inconsistent with implementation documentation. | | FAIL | |

The testing process has identified critical areas that require refactoring and corrections to ensure the reliability and functionality of the code. Further development will focus on addressing the failures and updating the code accordingly.

# White Box Test Implementation (MS04 Submission)

whitebox\_mapping\_tester.c

White Box testing below is added to the source file folder C file named as whitebox\_mapping\_tester.c

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Description** | **Test Data** | | **Expected Result** | | **Actual Result** | | **Pass/Fail** | | | **test\_addPointToRouteIfNot()** |  |  |  | Expected: Point added to route if it doesn't already exist. | | Actual: Point was not added due to incorrect parameter type. | | FAIL | |  | | **test\_distance()** |  | A math equation with colorful numbers and symbols  Description automatically generated with medium confidence |  | Expected: Distance should be 5.0 (sqrt(3^2 + 4^2)). | | Actual: Distance calculation incorrect due to potential overflow. | | FAIL | |  | | **test\_routeSizeLimitHandling()** |  | A black background with white text  Description automatically generated |  | Expected: Should prevent adding more points than MAX\_ROUTE. | | Actual: No overflow handling; point added despite exceeding limit. | | FAIL | |  | | **test\_shortestPathDocumentation()** |  |  |  | Expected: Function should execute without errors and match documented behavior. | | Actual: Function did not execute as per documentation; inconsistencies found. | | FAIL | | **test\_processInputErrorHandling()** |  |  |  | Expected: Should return specific error code based on conditions. | | Actual: Returned 0; no specific error code was provided. | | FAIL | |  |

As part of our MS04 submission, we focused on developing white box test cases for the internal functions implemented in finder.c. Given our understanding of the initial program and the further developments made, we anticipated a successful outcome for our tests.

While most of the tests passed successfully, the hasDestination() function did not produce the expected results. We suspect that this failure may be due to incorrect test data or a potential internal issue within the finder.c implementation itself.

A ticket has been created on the matrix for this failure (DEF3) and assigned to a team member for further debugging. We will continue to investigate the problem to ensure the reliability of the functionality and make necessary adjustments. Additionally, a Jira issue has been created for tracking purposes and further reporting on the resolution of this issue.