TEST DESCRIPTION

**TEST NAME OR ID**: T1-T4

**Test Type**: White box

**Description**: Test the internal logic of the isTruckCapacitySufficient function, including loop iterations, conditional paths, and capacity updates.

**Setup:** VS code unit test template, functions and a main() to execute the test function.

**Test Function**: isTruckCapacitySufficient

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  | | --- | | **Truck with Sufficient Capacity** |  |  | | --- | |  | | |  | | --- | | truck = {1, 3000, 120, 0, 0, {0}, nullptr}  package = {10, 5, {10, 10}} |  |  | | --- | |  | | 1 | 1 | Pass |
| |  | | --- | | **Truck with Insufficient Weight Capacity** |  |  | | --- | |  | | |  | | --- | | truck = {1, 100, 120, 0, 0, {0}, nullptr}  package = {150, 5, {10, 10}} |  |  | | --- | |  | | 0 | 0 | Pass |
| |  | | --- | | **Truck with Insufficient Volume Capacity** |  |  | | --- | |  | | truck = {1, 3000, 20, 0, 0, {0}, nullptr} package = {10, 25, {10, 10}} | 0 | 0 | PASS |
| |  | | --- | | **Truck with Insufficient Volume Capacity** |  |  | | --- | |  | | |  | | --- | | truck = {1, 100, 50, 0, 0, {0}, nullptr} package = {100, 50, {10, 10}} |  |  | | --- | |  | | 1 | 1 | PASS |

**TEST NAME OR ID**: T5-T8

**Test Type**: White box

**Description**: Test the internal logic of the validateShipment function, including loop iterations, conditional paths, and updates.

**Setup:**

**Test Function**: ValidateShipment

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Valid shipment** |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | shipment = { 10, 1, {10, 10} } |  |  | | --- | |  | | 1 | 1 | PASS |
| |  | | --- | | **Shipment with invalid size** |  |  | | --- | |  | | shipment = { 10, 2, {10, 10} } | 0 | 0 | PASS |
| |  | | --- | | **Shipment with invalid weight** |  |  | | --- | |  | | |  | | --- | | shipment = { -5, 1, {10, 10} } |  |  | | --- | |  | | 0 | 0 | PASS |
| |  | | --- | | **Shipment with invalid destination row (too high)** |  |  | | --- | |  | | |  | | --- | | shipment = { 10, 1, {26, 10} } |  |  | | --- | |  | | 0 | 0 | PASS |

**TEST NAME OR ID**: T9-T12

**Test Type**: White box

**Description**: Test the internal logic of the updateTruckCapacity function, including loop iterations, conditional paths, and updates.

**Setup:** VS code unit test template, functions and a main() to execute the test function.

**Test Function**: updateTruckCapacity

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  | | --- | | **Valid shipment updates truck capacity** |  |  | | --- | |  | | |  | | --- | | truck = { 1, 1000, 500, 0, 0, {0}, NULL }, shipment = { 100, 50, {10, 10} } |  |  | | --- | |  | | 1 | 1 | PASS |
| |  | | --- | | **Null truck pointer** |  |  | | --- | |  | | |  | | --- | | truck = NULL, shipment = { 100, 50, {10, 10} } |  |  | | --- | |  | | 0 | 0 | PASS |
| |  | | --- | | **Null shipment pointer** |  |  | | --- | |  | | |  | | --- | | truck = { 1, 1000, 500, 0, 0, {0}, NULL }, shipment = NULL |  |  | | --- | |  | | 0 | 0 | PASS |
| |  | | --- | | **Valid shipment causes negative capacity** |  |  | | --- | |  | | |  | | --- | | truck = { 1, 100, 50, 0, 0, {0}, NULL }, shipment = { 200, 100, {10, 10} } |  |  | | --- | |  | | 0 | 0 | PASS |

**TEST NAME OR ID**: T13-T16

**Test Type**: White box

**Description**: Test the internal logic of the checkPoint function, including loop iterations, conditional paths, and updates.

**Setup:** VS code unit test template, functions and a main() to execute the test function.

**Test Function**: checkPoint

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Verify that the function correctly identifies an adjacent point.** |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Route: [{1,1}, {2,2}], Destination: {2,1} |  |  | | --- | |  | | 0 | 0 | PASS |
| |  | | --- | | **Ensure the function returns -1 when no adjacent points are found.** |  |  | | --- | |  | | |  | | --- | | Route: [{1,1}, {2,2}], Destination: {4,4} |  |  | | --- | |  | | -1 | -1 | PASS |
| |  | | --- | | **Check the function's behavior when the route contains only one point.** |  |  | | --- | |  | | |  | | --- | | Route: [{1,1}], Destination: {1,2} |  |  | | --- | |  | | 0 | 0 | PASS |
| |  | | --- | | **Validate the function's behavior when the route is empty.** |  |  | | --- | |  | | |  | | --- | | Route: [], Destination: {1,1} |  |  | | --- | |  | | -1 | -1 | PASS |

**TEST NAME OR ID**: 17-20

**Test Type**: Whitebox

**Description**: Test that the function correctly validates initialization of package for a given shipment..

**Setup:** Initialize a PackageInfo struct and pass valid parameters.

**Test Function**: initializePackage

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Initialize with valid weight and size within range | |  | | --- | | Weight: 500.0, Size: 3.0, Destination: (10, 5) |  |  | | --- | |  | | 1 | 1 | PASS |
| |  | | --- | | **Initialize with negative weight** |  |  | | --- | |  | | |  | | --- | | Weight: -100.0, Size: 3.0, Destination: (10, 5) |  |  | | --- | |  | | 0 | 0 | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Initialize with destination row out of range** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Weight: 500.0, Size: 3.0, Destination: (26, 5) |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | 0 |  |  | | --- | |  | | 0 | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Initialize with valid weight and size at boundary location** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Weight: 500.0, Size: 3.0, Destination: (25, 25) |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | 1 |  |  | | --- | |  | | 1 | PASS |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**TEST NAME OR ID**: T21-24

**Test Type**: Whitebox

**Description**: Evaluates whether a given package can be accommodated by a specified truck based on its available weight and volume capacities

**Setup:** Initialize a Truck struct and pass valid parameters

**Test Function**: canFit

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  | | --- | | **Package fits within available weight and volume** |  |  | | --- | |  | | |  | | --- | | Package: weight=100, size=10, Truck: availableWeight=2000, availableVolume=40 |  |  | | --- | |  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  |   1 | 1 | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Package exceeds available weight** |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Package: weight=3000, size=20, Truck: availableWeight=2000, availableVolume=50 |  |  | | --- | |  | | |  | | --- | |  |   0 | 0 | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Package exceeds available weight** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Package: weight=200, size=5, Truck: availableWeight=2300, availableVolume=45 |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  |   1 | 1 | PASS |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Package exceeds both available weight and volume** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Package: weight=100, size=10, Truck: availableWeight=2400, availableVolume=40 |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  |   0 | 0 | PASS |

**TEST NAME OR ID**: T25-28

**Test Type**: Whitebox

**Description**: Compares the remaining capacities of two trucks to determine which one has more capacity available for use..

**Setup:** initialize truck1 (struct Truck\*): Pointer to the first truck whose remaining capacity is to be compared.

truck2 (struct Truck\*): Pointer to the second truck whose remaining capacity is to be compared

**Test Function**: compareRemaining

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  | | --- | | **Trucks with equal remaining weight should be considered equal.Hence any can be ranked higher** |  |  | | --- | |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Truck 1:** ID: 1000, Available Weight: 500, Available Volume: 50  **Truck 2:** ID: 1000, Available Weight: 500, Available Volume: 50 |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 1 |  |  | | --- | |  | |  |  | | --- | |  | | 1 | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Truck 1 with more remaining weight capacity should be ranked higher.** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Truck 1:** ID: 1000, Available Weight: 1000, Available Volume: 50  **Truck 2:** ID: 1000, Available Weight: 1500, Available Volume: 50 |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | -1 |  |  | | --- | |  | | -1 | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Truck 2 with more remaining weight capacity should be ranked higher.** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Truck 1:** ID: 1000, Available Weight: 1500, Available Volume: 50  **Truck 2:** ID: 1000, Available Weight: 1000, Available Volume: 50 |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | 1 | 1 | PASS |
| Truck 1 with more remaining volume capacity should be ranked higher when volume is the limiting factor. | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Truck 1:** ID: 1000, Available Weight: 1500, Available Volume: 50  **Truck 2:** ID: 1000, Available Weight: 1500, Available Volume: 75 |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | -1 |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | -1 | PASS |

**TEST NAME OR ID**: T29-32

**Test Type**: Whitebox

**Description**: Validate if weight or volume is the reason for not being able to initialize a a package

**Setup:** **Initialize a Truck and PackageInfo with valid data.**

**Test Function**: limitingFactor

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  | |  |  | | --- | |  | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | **Test weight as the limiting factor** |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **struct Truck truck = {**  **1, // truckID (example value)**  **2000, // availableWeight**  **30, // availableVolume**  **0, // allocatedShipments**  **0, // packageCount**  **{'B'}, // route**  **nullptr // packages (correctly initialized as nullptr)**  **};** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | W |  |  | | --- | |  | | | W | PASS |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Test volume as the limiting factor** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **struct Truck truck = {**  **1, // truckID (example value)**  **2000, // availableWeight**  **30, // availableVolume**  **0, // allocatedShipments**  **0, // packageCount**  **{'B'}, // route**  **nullptr // packages (correctly initialized as nullptr)**  **};** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | V |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | V | | PASS |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Test weight and volume when they are same** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **struct Truck truck = {**  **1, // truckID (example value)**  **2000, // availableWeight**  **30, // availableVolume**  **0, // allocatedShipments**  **0, // packageCount**  **{'B'}, // route**  **nullptr // packages (correctly initialized as nullptr)**  **};** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | V |  |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | V | | PASS |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Test weight and volume at 100 percent** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | **struct Truck truck = {**  **1, // truckID (example value)**  **2000, // availableWeight**  **30, // availableVolume**  **0, // allocatedShipments**  **0, // packageCount**  **{'B'}, // route**  **nullptr // packages (correctly initialized as nullptr)**  **};** |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | V | 1 |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | V | | PASS |