# Test Description

**Test Name or ID**: 1

**Test Type**: Black box

**Description**: Verifies truck's percentage weight capacity calculation works

**Setup:** Create Truck struct with different currentWeight values.

**Test Function**: percentageWeightFull

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Empty Truck | weight = 0 | 0.0 | 0.0 | Pass |
| Half Full | weight = 2500 | 0.5 | 0.5 | Pass |
| Full Capacity | weight = 5000 | 1.0 | 1.0 | Pass |
| Over Capacity | weight = 6000 | 1.0 | 1.0 | Pass |
| Empty Truck | weight = 0 | 0.0 | 0.0 | Pass |
| Half Full | weight = 2500 | 0.5 | 0.5 | Pass |
| Full Capacity | weight = 5000 | 1.0 | 1.0 | Pass |

**Bugs Found**:

None

**Test Name or ID**: 2

**Test Type**: Black box

**Description**: Verifies truck's volume percentage calculation is accurate.

**Setup:** Create Truck struct with different currentVolume values.

**Test Function**: percentageVolumeFull

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Empty Truck | volume = 0 | 0.0 | 0.0 | Pass |
| Half Full | volume = 100 | 0.5 | 0.48 | Fail |
| Full Capacity | volume = 200 | 1.0 | 1.0 | Pass |
| Over Capacity | volume = 250 | 1.0 | 1.0 | Pass |

**Bugs Found**:

percentageWeightFull function may be using integer division or incorrect rounding.

**Fixed**: Check if currentWeight / maxWeight uses floating-point math.

**Test Name or ID**: 3

**Test Type**: Black box

**Description**: Ensures that only valid delivery destinations are accepted.

**Setup:**

**Test Function**: validateDestination

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Valid Destination | map[0][0] = ' ', dest = (0,0) | true | true | Pass |
| On Building | map[10][10] = 'B', dest = (10,10) | false | false | Pass |
| Negative Index | dest = (-1,5) | false | false | Pass |
| Out of Bounds | dest = (25,25) | false | false | Pass |

**Bugs Found**:

None

**Test Name or ID**: 4

**Test Type**: Black box

**Description**: Determines the correct truck for a shipment based on capacity and proximity.

**Setup:** Create Truck array with mock values and use known shipment input.

**Test Function**: findTruckForShipment

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Valid Shipment | shipment(1000kg, 50m3, near blue) | 0 | 0 | Pass |
| Overweight Shipment | shipment(6000kg, 50m3) | -1 | -1 | Pass |
| First Truck Full | truck[0] full, shipment still fits in truck[1] | 1 | 1 | Pass |
| All Trucks Far | shipment far, use truck[2] | 2 | 2 | Pass |

**Bugs Found**:

None

**Test Name or ID**: 5

**Test Type**: Black box

**Description**: Calculates Manhattan distance between two valid points on a 25x25 map

**Setup:**

**Test Function**: calculateRouteDistance()

**Test Scenarios:**

Input: map[25][25] (not used in current logic)

Distance = abs(start.row - end.row) + abs(start.col - end.col)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Horizontal movement only | start = (3,3), end = (3,6) | 3 | 3 | Pass |
| Vertical movement only | start = (4,1), end = (7,1) | 3 | 3 | Pass |
| Diagonal movement | start = (2,2), end = (5,5) | 6 | 6 | Pass |
| Same start and end | start = (0,0), end = (0,0) | 0 | 0 | Pass |

**Bugs Found**:

None