# 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 |  |  |
| Half Full | weight = 2500 | 0.5 |  |  |
| Full Capacity | weight = 5000 | 1.0 |  |  |
| Over Capacity | weight = 6000 | 1.0 |  |  |
| Empty Truck | weight = 0 | 0.0 |  |  |
| Half Full | weight = 2500 | 0.5 |  |  |
| Full Capacity | weight = 5000 | 1.0 |  |  |

**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 |  |  |
| Half Full | volume = 100 | 0.5 |  |  |
| Full Capacity | volume = 200 | 1.0 |  |  |
| Over Capacity | volume = 250 | 1.0 |  |  |

**Bugs Found**:

None

**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 |  |  |
| On Building | map[10][10] = 'B', dest = (10,10) | false |  |  |
| Negative Index | dest = (-1,5) | false |  |  |
| Out of Bounds | dest = (25,25) | false |  |  |

**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 |  |  |
| Overweight Shipment | shipment(6000kg, 50m3) | -1 |  |  |
| First Truck Full | truck[0] full, shipment still fits in truck[1] | 1 |  |  |
| All Trucks Far | shipment far, use truck[2] | 2 |  |  |

**Bugs Found**:

None

**Test Name or ID**: 5

**Test Type**: Black box

**Description**: Finds the shortest valid path from a starting point to a destination

**Setup:**

**Test Function**: shortestPath

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Direct move | start = (2,2), dest = (2,3); clear path | Route has 1 point: {(2,3)} |  |  |
| Detour around building | start = (2,2), dest = (2,4); map[2][3] = 'B' | Route avoids (2,3) via adjacent squares |  |  |
| Destination blocked | dest = (5,5); All 8 neighbors of (5,5) set to 'B' | Route is empty |  |  |
| Same start and dest | start = (4,4), dest = (4,4) | Route has 0 points |  |  |

**Bugs Found**:

None