# Test Description

**Test Name or ID**: TI01

**Test Type**: Integration test

**Description**:

**Setup:**

**Test Function**: loadTruck + isTruckFull

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

Description of each bug found above and how to reproduce it.

**Test Name or ID**: TI03

**Test Type**: Integration test

**Description**:

**Setup:**

**Test Function**: read()

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Check if it says invalid destination | 100 0.5 3x | Invalid destination | Invalid destination | Pass |
| Check if it says invalid size | 100 2 2B | Invalid size | Invalid Size | Pass |
| Checks if weight is invalid | 1005 0.5 3X | Invalid weight entered, must be between 1-1000kg! | Invalid weight entered, must be between 1-1000kg! | Pass |
| Nominal test if all parameters are good | 1000 0.5 2B | struct Shipment package = {1000,0.5,{1,1}} | Struct Shipment package = {1000,0.5,{1,1}} | Pass |
| Exit condition | 0 0 x | Exits program | Ended up displaying a truck route for a package | Fail |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

Description of each bug found above and how to reproduce it.

**Test Name or ID**: TI02

**Test Type**: Integration test

**Description**:

**Setup:**

**Test Function**: mainProgram

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
|  |  |  | Ship on GREEN LINE,  divert:1A, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 1D, 2D, 3D, 4D, 4C, 4B, 1B, 1C, 14N |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

Description of each bug found above and how to reproduce it.

**Test Name or ID**: TI04

**Test Type**: Integration test

**Description**: Checking if the shortest path algorithm works correctly as expected

**Setup:** Under the visual studio 2022 environment.

**Test Function**: shortestPath

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Basic testing with shortestPath to see if it properly returns path | Point start(2,A)  Point dest(10, C) | Return a shortest path route that is non-zero num of points | Returned a shortest path route that is non-zero num of points | Pass |
| Testing edge of the map location case with shortest path | Point start(10,W)  Point dest(7,Y) | Returns a shortest path route that is a non-zero sum of points | Program crashes and goes out of bounds with index for array, adds too many points | Fail |
| Testing shortest path when it uses a building point as the destination point | Point start(9,W)  Point dest(8,Y) | Returns a shortest path route that is a non-zero sum of points | Program crashes and goes out of bounds with index for array, adds too many points | Fail |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

It gets stuck on the corners of the map or if a point is directly on top of a building being a 1 square and ends up adding too many points to the route by repeatedly going back and fourth on the same points. Crashing the array by going out of bounds as a result.