# Test Description

**Test Name or ID**: **F01T09-T014**

(Testing box weight)

**Test Type**: WhiteBox

**Description:** Validating the weight

**Setup:** debugger in vscode

**Test Function**: isValidWeight (int);

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| F01T09  Test with weight 500 (valid) | 500 | 1 (Valid) | 1 | 1 |
| F01T10  Test with weight 0 (invalid). | 0 | 0(Invalid) | 0 | Test is Pass |
| F01T11  Test with weight 2001 (invalid). | 2001 | 0(Invalid) | 0 | Pass |
| F01T12  Test with weight 1 (valid). | 1 | 1 (Valid) | 1 | Pass |
| F01T13  Test with weight -50 (invalid) | -50 | 0(Invalid) | 0 | Pass |
| F01T14  Test with weight 1000 (valid) | 1000 | 0(Invalid) | 0 | Pass |
|  |  |  |  |  |

**Bugs Found**:No bug Found

# Test 2

**Test Name or ID**: **F02T01-T07**

( Testing Size of box)

**Test Type**:Black box and Whitebox

**Description**: is the size valid or not.

**Setup:** debugger in vscode

**Test Function**: int isValidSize(float size);

**Test Scenarios:**

| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| F02T08:  Test with size 1 (valid). | 1 | 1(Valid) | 1 | Pass |
| F02T09:  Test with size 3 (invalid). | 3 | 0(Invalid) | 0 | Pass |
| F02T10:  Test with size 0.3 (invalid). | 0.3 | 0(Invalid) | 0 | Pass |
| F02T11:  Test with size 0.5 (valid). | 0.5 | 1(Valid) | 1 | Pass |
| F02T12:  Test with size -9 (invalid). | -9 | 0(Invalid) | 0 | Pass |
| F02T13:  Test with size 4.5 (invalid). | 4.5 | 0(Invalid) | 0 | Pass |

**Bugs Found**:No bugs found

# Test 3

**Test Name or ID**: **F03T01-T08**

(Testing destination)

**Test Type**: Whitebox

**Description**: is the destination valid or not.

**Setup:** debugger in vscode

**Test Function**: isvalidDestination;

**Test Scenarios:**

| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| F03T09:  Test with destination "123A" (valid). | “123A” | 1(valid) | 1 | pass |
| F03T10:  Test with destination "1234" (invalid). | “1234” | 0(invalid) | 0 | pass |
| F03T11:  Test with destination "123A4" (invalid). | “123A4” | 0(invalid) | 0 | pass |
| F03T12:  Test with destination "1A" (valid). | “1A” | 1(valid) | 1 | pass |
| F03T13:  Test with destination "12y" (valid). | “12y” | 0(invalid) | 0 | pass |
| F03T14:  Test with destination "1@" (invalid). | “1@” | 0(invalid) | 0 | pass |

**Bugs Found**:No bug found

# Test 4

**Test Name or ID:**  **F04T07-T012**

**(**Testing determineDiversion)

**Test Type**: Whitebox

**Description:**

**Setup:** debugger in vscode

**Test Function**: determineDiversion

**Test Scenarios:**

| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| F04T07 :  Test with weight 1800 and destination "100A" (no diversion). | Weight= 1800  Destination = 100A | Weight < 1900, no diversions generated. | Weight < 1900, no diversions generated. | Pass |
| F04T08 :  Test with weight 1900 and destination "2D" (valid diversions). | Weight = 1900  Destination = 2D | Starting determineDiversion with weight: 1900, destination: 2D  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 2, base\_letter = D  Generated diversion: 0D  Generated diversion: 1D  Generated additional diversion with previous letter: 2A  Generated additional diversion with previous letter: 2B  Generated additional diversion with previous letter: 2C  Total diversions generated: 13 | Starting determineDiversion with weight: 1900, destination: 2D  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 2, base\_letter = D  Generated diversion: -8D  Generated diversion: -7D  Generated diversion: -6D  Generated diversion: -5D  Generated diversion: -4D  Generated diversion: -3D  Generated diversion: -2D  Generated diversion: -1D  Generated diversion: 0D  Generated diversion: 1D  Generated additional diversion with previous letter: 2A  Generated additional diversion with previous letter: 2B  Generated additional diversion with previous letter: 2C  Total diversions generated: 13 | FAIL |
| F04T09 :  Test with weight 2500 and destination "7B" (valid diversions). | Weight = 2500  Destination = 7B | Starting determineDiversion with weight: 2500, destination: 7B  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 7, base\_letter = B  Generated diversion: 0B  Generated diversion: 1B  Generated diversion: 2B  Generated diversion: 3B  Generated diversion: 4B  Generated diversion: 5B  Generated diversion: 6B  Generated additional diversion with previous letter: 7A  Total diversions generated: 11 | Starting determineDiversion with weight: 2500, destination: 7B  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 7, base\_letter = B  Generated diversion: -3B  Generated diversion: -2B  Generated diversion: -1B  Generated diversion: 0B  Generated diversion: 1B  Generated diversion: 2B  Generated diversion: 3B  Generated diversion: 4B  Generated diversion: 5B  Generated diversion: 6B  Generated additional diversion with previous letter: 7A  Total diversions generated: 11 | FAIL |
| F04T10 : Test with weight 200 and destination "2A" (no diversion). | Weight = 200  Destination = “2A” | Starting determineDiversion with weight: 200, destination: 2A  Weight < 1900, no diversions generated. | Starting determineDiversion with weight: 200, destination: 2A  Weight < 1900, no diversions generated. | pass |
| F04T11 :  Test with weight 2000 and destination "25A" (valid diversions). | Weight = 2000  Destination = “25A” | Starting determineDiversion with weight: 2000, destination: 25A  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 25, base\_letter = A  Generated diversion: 15A  Generated diversion: 16A  Generated diversion: 17A  Generated diversion: 18A  Generated diversion: 19A  Generated diversion: 20A  Generated diversion: 21A  Generated diversion: 22A  Generated diversion: 23A  Generated diversion: 24A  Total diversions generated: 10 | Starting determineDiversion with weight: 2000, destination: 25A  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 25, base\_letter = A  Generated diversion: 15A  Generated diversion: 16A  Generated diversion: 17A  Generated diversion: 18A  Generated diversion: 19A  Generated diversion: 20A  Generated diversion: 21A  Generated diversion: 22A  Generated diversion: 23A  Generated diversion: 24A  Total diversions generated: 10 | pass |
| F04T12 :  Test with weight 2500 and destination "22T" (Failed) | Weight = 2500  destination = “22T” | Starting determineDiversion with weight: 2500, destination: 22T  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 22, base\_letter = T  Generated diversion: 12T  Generated diversion: 13T  Generated diversion: 14T  Generated diversion: 15T  Generated diversion: 16T  Generated diversion: 17T  Generated diversion: 18T  Generated diversion: 19T  Generated diversion: 20T  Generated diversion: 21T  Generated additional diversion with previous letter: 22A  Generated additional diversion with previous letter: 22B  Generated additional diversion with previous letter: 22C  Generated additional diversion with previous letter: 22D  Generated additional diversion with previous letter: 22E  Generated additional diversion with previous letter: 22F  Generated additional diversion with previous letter: 22G  Generated additional diversion with previous letter: 22H  Generated additional diversion with previous letter: 22I  Generated additional diversion with previous letter: 22J  Generated additional diversion with previous letter: 22K  Generated additional diversion with previous letter: 22L  Generated additional diversion with previous letter: 22M  Generated additional diversion with previous letter: 22N  Generated additional diversion with previous letter: 22O  Generated additional diversion with previous letter: 22P  Generated additional diversion with previous letter:  Generated additional diversion with previous letter: 22R  Generated additional diversion with previous letter: 22S  Total diversions generated: 84 | Starting determineDiversion with weight: 2500, destination: 22T  Weight >= 1900, proceeding with diversion calculation.  Parsed destination: base\_number = 22, base\_letter = T  Generated diversion: 12T  Generated diversion: 13T  Generated diversion: 14T  Generated diversion: 15T  Generated diversion: 16T  Generated diversion: 17T  Generated diversion: 18T  Generated diversion: 19T  Generated diversion: 20T  Generated diversion: 21T  Generated additional diversion with previous letter: 22A  Generated additional diversion with previous letter: 22B  Generated additional diversion with previous letter: 22C  Generated additional diversion with previous letter: 22D  Generated additional diversion with previous letter: 22E  Generated additional diversion with previous letter: 22F  Generated additional diversion with previous letter: 22G  Generated additional diversion with previous letter: 22H  Generated additional diversion with previous letter: 22I  Generated additional diversion with previous letter: 22J  Generated additional diversion with previous letter: 22K  Generated additional diversion with previous letter: 22L  Generated additional diversion with previous letter: 22M  Generated additional diversion with previous letter: 22N  Generated additional diversion with previous letter: 22O  Generated additional diversion with previous letter: 22P  Generated additional diversion with previous letter:  Generated additional diversion with previous letter: 22R  Generated additional diversion with previous letter: 22S  Total diversions generated: 84 | Fail |

**Bug: letter Z is not in the diagram and number is until 25. It might need to be fixed.**

# Test 5

**Test Name or ID:**  **F05T08-T013**

**(**Testing Remaining Capacity)

**Test Type**: Whitebox

**Description:** what is the remaining capacity.

**Setup:** debugger in vscode

**Test Function**: int calculateRemainingCapacity ( int currentWeight, float currentVolume);

**Test Scenarios:**

| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| F05T08:Test with weight 0 and volume 0 (full capacity remaining). | currentWeight=0  currentVolume = 0.0 | Calculating remaining capacity with currentWeight = 0 and currentVolume = 0.  Remaining weight: 2000 kg, Remaining volume: 20.00 cubic meters | Calculating remaining capacity with currentWeight = 0 and currentVolume = 0.  Remaining weight: 2000 kg, Remaining volume: 20.00 cubic meters | pass |
| F05T09: Test with weight 2100 and volume 21 (remaining capped at zero). | currentWeight= 2100  currentVolume = 21.0 | Calculating remaining capacity with currentWeight = 2100 and currentVolume = 21.  Remaining weight went below zero, setting to zero.  Remaining volume went below zero, setting to zero.  Remaining weight: 0 kg, Remaining volume: 0.00 cubic meters | Calculating remaining capacity with currentWeight = 2100 and currentVolume = 21.  Remaining weight went below zero, setting to zero.  Remaining volume went below zero, setting to zero.  Remaining weight: 0 kg, Remaining volume: 0.00 cubic meters | pass |
| F05T10: Test with weight 1000 and volume 10 (half capacity remaining). | currentWeight= 1000  current Volume = 10.0 | Calculating remaining capacity with currentWeight = 1000 and currentVolume = 10.  Remaining weight: 1000 kg, Remaining volume: 10.00 cubic meters | Calculating remaining capacity with currentWeight = 1000 and currentVolume = 10.  Remaining weight: 1000 kg, Remaining volume: 10.00 cubic meters | pass |
| F05T11:Test with weight -500 and volume -5 (full capacity remaining). | currentWeight= -500  currentVolume = -5.0 | Calculating remaining capacity with currentWeight = -500 and currentVolume = -5.  Remaining weight: 2500 kg, Remaining volume: 25.00 cubic meters | Calculating remaining capacity with currentWeight = -500 and currentVolume = -5.  Remaining weight: 2500 kg, Remaining volume: 25.00 cubic meters | pass |
| F05T12: Test with weight 1500 and volume 25 (remaining volume capped at zero). | currentWeight= 1500  currentVolume = 25.0 | Calculating remaining capacity with currentWeight = 1500 and currentVolume = 25.  Remaining volume went below zero, setting to zero.  Remaining weight: 500 kg, Remaining volume: 0.00 cubic meters | Calculating remaining capacity with currentWeight = 1500 and currentVolume = 25.  Remaining volume went below zero, setting to zero.  Remaining weight: 500 kg, Remaining volume: 0.00 cubic meters | pass |
| F05T13: Test with weight 2000 and volume 10 (valid remaining capacity). | currentWeight= 2000  currentVolume = 10.0 | Calculating remaining capacity with currentWeight = 2000 and currentVolume = 10.  Remaining weight: 0 kg, Remaining volume: 10.00 cubic meters | Calculating remaining capacity with currentWeight = 2000 and currentVolume = 10.  Remaining weight: 0 kg, Remaining volume: 10.00 cubic meters | pass |

**Bugs Found**:No