# Function Description

**Function Name:** checkWeight

**Parameter List: int weight**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| weight | Int | Weight The shipment weight in kilograms. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Return 1 if the weight is valid; otherwise, 0

**Description:** Checks if the shipment weight is within acceptable limits (1 to TRUCK\_MAX\_WEIGHT).

**Function Name:** checkBoxSize

**Parameter List: double boxSize**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| boxSize | double | Allowed box sizes are 0.5, 2, and 5 cubic meters. boxSize The size of the box in cubic meters. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return 1 if the box size is valid; otherwise, 0.

**Description:** Checks if the box size is valid.

**Function Name:** isValidDestinationFormat

**Parameter List: const char\* dest**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| dest | const char\* | dest The destination string provided by the user. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Return 1 if the format is valid; otherwise, 0.

**Description:** Validates the format of the destination string. The destination should consist of a numeric part (0-25) followed by an uppercase letter (A-Y). For example: "12L" is valid, while "28x" or "3Z" are invalid.

**Function Name:** checkDestination

**Parameter List: const char\* dest, const struct Map\* map**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| dest | const char\* | dest The destination string provided by the user. A pointer to the current map. |
| map | const struct Map\* | A pointer to the current map. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return 1 if the destination is valid; otherwise, 0.

**Description:** Checks if the destination input is valid. First, it validates the format (using isValidDestinationFormat). Then, it converts the destination into map coordinates and checks whether the destination corresponds to a valid building location on the map.