# Function Description

**Function Name:** isPackageValid

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| weight | double | the weight of the package in kg |
| volume | double | the volume of the package in m3 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return 1 if the package is valid, 0 if it is not

**Description:** Checks if a packages weight and volume are valid:

* The **weight** must be within the range of 1 to 1000 kilograms.
* The **volume** must be one of the defined sizes (SMALL\_BOX, MEDIUM\_BOX, LARGE\_BOX).

**Function Name:** isDestValid

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| map | const struct Map\* | the map to query |
| dest | struct Point | the destination point |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return 1 if the destination is valid, 0 if it is not

**Description:** This function checks if a given destination on the map is valid. A destination is considered valid if:

* The row and col of the destination are within the map's bounds.
* The square at the destination does not represent a building.

**Function Name:** addPackage

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| truck | struct Truck\* | The truck to add the package to |
| package | const struct Package\* | The package to be added |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** no return value.

**Description:** This function adds a package to a truck

* Adds the package to the truck's packages array at the next available index
* Increments the truck's number of packages
* Updates the truck’s weight and volume.

**Function Name:** isPackageAcceptable

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| truck | const struct Truck\* | The truck to check |
| package | const struct Package\* | The package to check |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return 1 if the package can be acceptable, 0 if it cannot

**Description:** This function checks if a package can be acceptable by a truck without exceeding its weight or volume limit:

**Function Name:** calcDeliveryPath

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| map | const struct Map\* | The map to query |
| route | const struct Route\* | The truck's route |
| point | const struct Point\* | The destination point |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return a delivery structure with the details about the delivery

**Description:** This function calculates the shortest delivery path for a package. It performs the following steps:

1. Iterates through the points on the truck's route to find the one closest to the destination using **Euclidean distance**.
2. Calculates the shortest path from this closest point to the destination using the **A**\* algorithm, considering buildings as obstacles.
3. Returns a Delivery structure containing:

* closestPoint: The nearest route point to the destination.
* routeDist: The Euclidean distance from the closest point to the destination.
* path: The shortest path to the destination as a sequence of points.
* isPossible: 1 if the path exists, otherwise 0

**Function Name:** getNumRows

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| map | const struct Map\* | the map to query |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return the number of rows in the map.

**Description:** This function retrieves the total number of rows in a given map.

**Function Name:** getNumCols

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| map | const struct Map\* | the map to query |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** return the number of columns in the map

**Description:** This function retrieves the total number of columns in a given map.

**Function Name:** printMap

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| map | const struct Map\* | the map to query |
| base1 | int | if true print row indices from 1 up otherwise 0 up |
| alphaCols | int | if true print col header as letters, otherwise numbers |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** no return value.

**Description:** Prints a visual representation of the map with row and column labels. Columns can be labeled alphabetically or numerically, depending on the value of alphaCols. The function uses predefined symbols (sym) to represent different map elements.

**Function Name:** getBlueRoute

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
|  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** the route for the blue trucks.

**Description:** Build and return the route for the blue trucks.

**Function Name:** getGreenRoute

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** the route for the green trucks.

**Description:** Build and return the route for the green trucks.

**Function Name:** getYellowRoute

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** the route for the yellow trucks.

**Description:** Build and return the route for the yellow trucks.