# Function Description

1. **Function Name:** **validTruckPaths()**

**Parameter List: shipment, truck, map, routes, size**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| shipment | struct | Shipment struct that holds the weight, volume and destination |
| truck | struct | Truck struct that holds the current cargo weight, volume and the route it takes |
| map | struct | Map struct will all the points in the map |
| route | array | Array of pointers to route structs to store the valid routes |
| size | int | Pointer to an integer holding the size of the ‘routes’ array |

**Returns:** void (updates the routes array through a pointer and size).

**Description:** Checks and identifies valid routes for a truck to deliver a shipment based on the given map.

1. **Function Name:** **checkDestination()**

**Parameter List: route, shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| route | struct | Pointer to a structure representing a route which is a collection of points |
| shipment | struct | Contains information about the shipment’s destination |

**Returns:** returns an int: 1 if the destination is in the route, 0 if the destination is not in the route

**Description:** Checks if the given shipment’s destination is within the specified route

1. **Function Name:** **routePoints()**

**Parameter List: route, shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| shipment | struct | Shipment struct that holds the weight, volume and destination |
| route | array | Structure holding a collection of points defining a route |

**Returns:** void, no return value

**Description:** Processes the points of a given route that a truck can travel to.

1. **Function Name:** **bestRoute()**

**Parameter List: routes, shipment, size**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| shipment | struct | Shipment struct that holds the weight, volume and destination |
| route | array | Array of pointers to route structs to store the valid routes |
| size | int | Pointer to an integer holding the size of the ‘routes’ array |

**Returns:** An integer that holds the index of the best route in the ‘routes’ array

**Description:** Determines and returns the index of the best (shortest or optimal) route to deliver the shipment.

1. **Function Name:** **validShipment()**

**Parameter List: shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| shipment | struct | Shipment struct that holds the weight, volume and destination |

**Returns:** An int: 1 if the shipment size is valid, 0 if the shipment is invalid

**Description:** Checks if the shipment's size is within valid limits based on the truck's restrictions.

1. **Function Name:** **checkBuilding()**

**Parameter List: route, shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| shipment | struct | Shipment struct that holds the weight, volume and destination |
| route | array | Structure holding a collection of points defining a route |

**Returns:** An int: 1 if the route passes through valid building locations, 0 if the route does not pass through valid building locations

**Description:** Verifies if the route passes through valid building locations based on the map.