# Function Description

**Function Name: struct Shipment read()**

**Parameter List: It does not receive any parameter.**

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

**Returns:** Returns a newly created shipment structure using the results of the user input

**Description:** This function is to prompt user to input the shipment weight, box size and destination. Also, it will use functions for validation routines – int validWeight, int validSize and validDesination to verify the result. If the validations all pass, a shipment struct would be created and returned.

**Function Name:** validWeight

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **weight** | double | The weight of a box |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns an integer which is 1 if weight is between 1 and 1000, 0 otherwise if weight isn't valid

**Description:** Checks if the user input for the weight of a shipment is within a valid range and is invalid if this returns 0

**Function Name:** validSize

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **size** | double | The size of the box |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns an integer that is 1 if it’s within the valid size range that is 0.25,0.5,1.0 respectively, 0 otherwise if it's not any of those

**Description:** Checks if the user input for the size of the shipment is within a valid range and is invalid if it returns 0

**Function Name:** validDestination

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **m** | struct Map | This is the map of the delivery area |
| **dest** | char \* | This is the destination for the delivery |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns integer, 1 if it is a valid destination square, 0 otherwise if it isn't valid

**Description:** Checks if the user input for the destination is valid which should be on a building square and is invalid if it returns 0

**Function Name:** isTruckFull

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **truck** | struct Truck | The truck being checked for space or weight |
| **box** | struct Shipment | The shipment being checked for space or weight |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** An integer is returned for determining space or weight. 1 if it is at maximum capacity and 0 otherwise

**Description:** Checks if the truck can actually carry the shipment by looking at weight and space and returns a 0 if it’s possible to put on the truck

**Function Name:** isOnOriginalPath

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **p** | struct Point \* | The point being checked |
| **origRoute** | struct Route \* | The original route of the truck |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** if this point is on the route is it 1 if it is a divergent point and 0 otherwise if it is on the path of the truck

**Description:** Checks if this point is on the original path of the truck and returns a 0 if it’s not on the path

**Function Name:** calcMinDistance

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **truck** | struct Truck \* | The truck being checked for min distance |
| **dest** | struct Point \* | The destination point for calculating every distance |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns the point that is the closest point to the destination for the truck’s route

**Description:** Calculate the minimum distance out of every point of the route for a truck and return the closest point on it’s route to the destination

**Function Name:** findIndexOfShortestDist

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **distanceList** | double[] | An array list that is the calculated distances of every point for the truck route |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns an integer that is the index for the shortest calculated distance in the truck

**Description:** Find shortest distance among elements in the calculated distance array for a truck route

**Function Name:** determineDivergencePath

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **truck** | struct Truck | The truck being used to store the divergence path |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Does not return anything but the divergence path is going to be stored in truck pointer’s divergence route variable

**Description:** Calculates the divergence path of truck by comparing the shortest path and the original route. Any points that do not match will be stored as deviance points.

**Function Name:** setValidDefaultMinTruck

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **truckList** | struct Truck[] | An array of trucks that is going to be used to set min |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Will return an integer index value of the first valid default min truck otherwise returns –1 if none of the trucks can reach the destination

**Description:** Loops through all trucks in truck array and tries to set a default minimum truck index for later use when it is comparing shortest distances to weed out any trucks that do not successfully deliver anything

**Function Name:** capPercentage

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **t** | struct Truck | The truck being used to calculate the total percentage |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Will return the total capacity percentage for comparison purposes

**Description:** Calculate total percentage of weight and space to decide which truck to load if they are the same distance from the delivery

**Function Name:** findBestTruckForShipment

**Parameter List:**

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Type** | **Description** |
| **m** | struct Map | Map of delivery space with buildings |
| **truckList** | struct Truck[] | The list of trucks for delivery |
| **package** | struct Shipment | The shipment for delivery |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns an integer that is the index for the truck that can do the shipment with the shortest path, otherwise return -1 if it can't find any trucks.

**Description:** Takes into consideration the capacity of the truck, shipment size and weight, route of the truck tries to place which is closest to destination. If no truck can take it, then returns -1