# Function Description

**Function Name:** checkSpace

**Parameter List: Truck truck, Shipment shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| truck | Truck | Stores the given truck to check if it has space |
| shipment | Shipment | The given shipment to store on a given a truck |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns true or false depending on the shipment (if the shipment is valid, or not) or based on the truck conditions (has enough space for package or not)

**Description:** This function simply checks if the package is valid, and if the given truck’s weight – subtracted by the weight of the package is valid, as well as the same check for the size(volume) of the package. If all returns true this function will return true.

**Function Name:** valid

**Parameter List: Shipment shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| shipment | Shipment | The given shipment to check if it’s within range (valid) |
| map | Map | Map to check if address is valid |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Returns a Boolean value, true or false depending on if the shipment is valid

**Description:** This function checks if the shipment is within range (1-5000kg) for weight, if the shipment volume is within range, and the address is an actual address corresponding to the map (runs a for loop to check or checks if the intersection exits via arr)

**Function Name:** assignPackage

**Parameter List: Map map, Truck trucks[], Shipment shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| Map | Map | Stores map |
| trucks | Array | Stores the total trucks and their states |
| shipment | Shipment |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** This function returns the index of the truck array that this package is assigned to if successful, it will -1 otherwise any error is encountered.

**Description:** This function will loop through all trucks, and first check if the current trucks route is within range (directly) of the route, if not it will run a secondary check to see if the truck has space, then move onto the next one. If any one of the trucks is on-route, that package will be assigned, otherwise the closest truck (with space) will be assigned, if a diversion is needed (not on-route) it will call the diversion function to handle the rest, and return the address of the current truck.

**Function Name:** Divert

**Parameter List: Map map, Truck trucks[], Shipment shipment**

|  |  |  |
| --- | --- | --- |
| Parameter Name | Type | Description |
| Map | Map | Stores map |
| trucks | Array | Stores the total trucks and their states |
| shipment | Shipment |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Returns:** Nothing, will be void only handle printing of diversion/state changes of data.

**Description: This function will handle diversion and be given relevant information about diverting a truck, using the shortest path function it will find the least amount of diversion from a given truck route to meet the point address of the shipment, it will then print the new diversion route.**