Function Description

Function Name: isValidWeight // int isValidWeight (struct Truck truck, struct Package package)

Parameter List:

Parameter Name	Type	Description
truck	struct	Here we check the capacityHolding and weight attribute of the truck struct in order to determine whether any more packages are eligible to be added onto the truck. The maximum a truck can hold is 48 cubic meters of boxes and the maximum weight holding capacity is 1500kg.
package	struct	Here we check the package weight and capacity so that we can find the total weight and capacity and determine whether the package can be added to the truck or not.

Returns: int

returns 1 if the total volume and weight is less than the maximum volume and weight the truck can handle so, the package is added.

returns 0 if the total volume and weight is more than the maximum volume and weight the truck can handle so the package is rejected.

Description: As part of our application, when a shipment arrives it has to be determined if the shipment can be loaded onto a particular truck and in order to determine that we have to determine if the total capacity and weight of the after adding the package and make sure that it is less than the maximum capacity which is 48 cubic meters and maximum weight which is 1500kg. To determine that this function will return an integer 1 which says that it can be loaded or integer 0 which says that the package cannot be loaded.

Function Description

Function Name: promptUser int promptUser(struct Package* package)

Parameter List:

Parameter Name	Type	Description
package	Struct Package	In this function we shall prompt a user for input and pass this input into the structure variables.

Returns: int

The function will return 1 when input was passed successfully and when the input is valid.

And return 0 when input is invalid

(Wish we could have Booleans)

Description :

As the program does not have prompting user function, I decided to implement one. There are limitations on cargo and destination that are present in the function. Capacity limitation: 48 cubic meters (passed as in ½ cubic meters, for example 3 cubic meters must entered as 1.5), Weight limitations: 1500 kilograms and of course the destination have to be in a building.

Function Description

Function Name: dispatchTruck

struct Truck *dispatchTruck (struct Truck* trucks, struct Route *routes, struct Point destination)

Parameter List:

Parameter Name	Type	Description
trucks	Struct Truck Pointer	Return a truck from this list, that can navigate to destination.
routes	Struct Route pointer	Use the routes that are available to accomplish the task of reaching the destination
destination	Struct Point	The final point where a truck needs to reach

Returns: struct Truck*

Returns a struck Truck pointer, with an updates route variable that will be going nearest to the destination provided as struct Point. The pointer will be null if there is no method to accomplish the delivery (ex: All trucks are full).

Description: The dispatchTruck function is the focal point of the project. It will return to a truck with a given route, that can complete the delivery. It will be using most of the other custom and premade functions to work. It will be taking in trucks array [as a pointer] and chose the best truck for the destination. It will be using routes to navigate for the most part. This function will also update the route of the returned Truck to match the one it should follow.

Function Description

Function Name: limitFactor

Parameter List: struct truckInfo

Parameter Name	Type	Description
truckInfo	struct	This is a structure that contains information about trucks such as weight, volume and limiting factor of that truck

Returns: The return type of this function is double as it will return the percentage of limiting factor. The special conditions on which the return value depends are size of the truck and volume.

Description: It is to find the limiting factor of each truck as it is the deciding condition for the most appropriate selection for truck. This function finds out the percentage of both the weight and volume, compares them and returns the greater between the two.

Function Description

Function Name: isTruckFull // int isTruckFull (struct Truck *truck)

Parameter List:

Parameter Name	Type	Description
truck	struct	Here we check the capacityHolding and weight attribute of the truck struct in order to determine whether any more packages are eligible to be added onto the truck. The maximum a truck can hold is 48 cubic meters of boxes and the maximum weight holding capacity is 1500kg.

Returns: Boolean

returns 1 if the truck is full

returns 0 if there still space in the truck

Description: As part of our application, when a package is place, we must determine if there is still space in the truck to put the package, this is defined by the size and weight of the package and the capacity for the trucks.

Function Description

Function Name: allTrucksAreFull | int allTrucksAreFull(struct Truck *trucks, int numTrucks)

Parameter List:

Parameter Name	Type	Description
truck	struct	Here we check the capacityHolding and weight attribute of the truck struct in order to determine whether any more packages are eligible to be added onto the truck. The maximum a truck can hold is 48 cubic meters of boxes and the maximum weight holding capacity is 1500kg.
numTrucks	int	The number of trucks available to iterate in the struckt

Returns: Boolean

returns 1 if all the trucks are full

returns 0 if there still space in one truck

Description: As part of our application, we have to ensure that there is still the availability of getting another package on any of the routes. This function is combined with the istruckFull to check all the trucks and their capacity to get another package.

Function Description

Function Name: printRoute | void printRoute(int truckIndex, struct Route *route)

Parameter List:

Parameter Name	Type	Description
truckIndex	Int	This varibale is the one that defines the color of the route that we are using if It is =0 is the blue, if =1 is the yellow, if equals other number equals to green.
Route	struct	This struck has the point where the trucks intersect, as well as the routeSymbol

Returns: Printf

"BLUE, YELLOW, GREEN, AND THE ROUTE"

Description: This function prints the route in the map bases in the index of the route that has been entered, also it draws the intersections of the route in the map based on the struct of the route.

Function Description

Function Name: getDistance | int getDistance(struct Point start, struct Point end)

Parameter List:

Parameter Name	Type	Description
Point	Struct	This struct contains to point the row and the column on the axis

Returns: struct

It return the struct with the number of point of the route.

Description: This function help us to calculate the distance between two points and the point of on the route with their distance of the objective.