- In the class car we initialized the brand, type, plate, speed, model
- in the public we created a constructor to set the default values for these variables
- Then we created setters and getters for every element to be able to call the function and retrieve the values sent from the main
- the set car type includes the validation to check this string we received from the main against private, motorcycle, truck, or bus
- In the class Rd we initialize the speed limit, road type, and the counters
- We create a constructor check out the road type and speed limit from the mean and set it as our variables in the class
- Then we create centers and gutters to be able to set a retrieve the information we got from the main as the variables we initialized in our class and the road typesetter acts as validation check whether it is character a or B or C before setting it
- The function radar Check the road type speed limit with the car speed limit and function allow checks which roads are available for the card type by incrementing the counters and function age returns the age of the car
- in the main we create the car objects and set they're in the moments and likewise for the road
- we then create a queue and push every car element in it Then we create a new object of car to check every first element of the Q and through it access the car type, model, speed,ect.
- Then we create if conditions regarding the radar with the road type and the car speed limit and if the condition is true and the car is above the speed limit then we print the information of the car then we proceed to pop the queue element
- lastly, we got the number of cars that each road has been allowed on from the get counters and calculate the efficiency of each road

Errors: the repetition of the Cout statements in the function allow and the efficiency calculation is incorrect. The efficiency calculator should work because when it couts get count it prints the correct output.