In this exercise we still study smart pointers (and threads).

Excercise 13 (Smart pointers and pointer containers, 4p)

Example program car_sales.cpp has a large number of memory leaks. Your task is to fix the memory leaks. Replace all raw pointers with smart pointers (shared_ptr / weak_ptr) so that there is no need to call delete anywhere in the program.

Modify class Website in the following way:

- Delete remove() member function from the class
- Modify print() so that it deletes automatically all sold cars before printing the list of advertised cars

Extra excercise (Smart pointers and pointer containers, 2p)

Add a (smart) pointer to Dealer in class Car so that when car information is printed the dealer name is also printed. Note that this creates circular reference!

Extra excercise (Threads, 2p)

Modify class Website so that it has member function run() that runs a loop that sleeps for 10 seconds and then prints a list of advertised cars. The loop, and thread, stops if the list of cars is empty.

Change the program so that after initial setup (filling of the lists) a thread is started for each website. Remove all calls to print functions from the main.

Change car_sales() so that it asks user to select the dealer from which to buy a car or to exit program in a loop. Call sell() of selected dealer or stop loop and wait for the other thread(s) to finish if user selected to exit program.