Homework #3

In this assignment you will be using quicksort to sort an array of car objects by various criteria.

Define a `struct Car` as follows:

```
#define MAX_STRING_LENGTH

typedef struct Car_ {
     char make[MAX_STRING_LENGTH];
     char model[MAX_STRING_LENGTH];
     int mpg;   /* Miles per gallon */
} Car;
```

a) **(2 points)** Implement a function called `compareCarsByMakeThenModel` that can be passed as an argument to the `compare` parameter of the `qksort` function from the book. `compareCarsByMakeThenModel` should return a value that will cause `qksort` to sort an array of cars in ascending order (from smallest to largest) by `make` and, when two cars have the same `make`, in ascending order by `model`.

b) **(2 points)** Implement a function called `compareCarsByDescendingMPG` that can be passed as an argument to the `compare` parameter of the `qksort` function from the book. `compareCarsByDescendingMPG` should return a value that will cause `qksort` to sort an array of cars in descending order (from largest to smallest) by `mpg`.

c) **(2 points)** Implement a function called `compareCarsByMakeThenDescendingMPG` that can be passed as an argument to the `compare` parameter of the `qksort` function from the book. `compareCarsByMakeThenDescendingMPG` should return a value that will cause `qksort` to sort an array of cars in ascending order by `make` and, when two cars have the same `make`, in descending order by `mpg`.

d) **(3 points)** Write a program that tests your functions from parts a-c with the following array of cars:

```
Car cars[] = {
    { "Toyota", "Camry", 33 },
    { "Ford", "Focus", 40 },
    { "Honda", "Accord", 34 },
    { "Ford", "Mustang", 31 },
    { "Honda", "Civic", 39 },
    { "Toyota", "Prius", 48 },
    { "Honda", "Fit", 35 },
    { "Toyota", "Corolla", 35 },
    { "Ford", "Taurus", 28 }
 };
```

Your test program should do the following:

1. Output (displaying make, model, and MPG) the cars in original unsorted order.
2. Output the cars sorted (using qksort from the book) by make then model.
3. Output the cars sorted (using qksort from the book) by descending MPG.
4. Output the cars sorted (using qksort from the book) by make then descending MPG.

e) **(1 point)** Make sure your source code is well-commented, consistently formatted, uses no magic numbers/values, follows programming best-practices, and is ANSI-compliant.

Turn in all source code, program output, diagrams, and answers to questions in a single Word or PDF document.