The aim of this task is to demonstrate your ability to solve problems using the functional programming language Haskell. You should show that you can:

1. use data type definitions to appropriately model the data being manipulated.
2. write functions to express the desired transformations.
3. use higher-order functions to economically express your solution.
4. use recursion where it is required.
5. structure your code so that it is easy to read and maintain.
6. recognize generalizations that make your solution flexible and extensible.

Write a program to do some sort of data analysis.

Most of these things can be done with the libraries shipped with GHC, but if you wish you may use Haskell packages published on the Hackage site, as long as they run on any system. Hackage packages can be installed with the Cabal tool bundled with GHC.)

Your program makes sophisticated use of Haskell features to produce a solution that is clean, clear and flexible, and demonstrates a sophisticated grasp of the possibilities of the language. The extra credit is not for more code or more features, but for the elegance and clarity of your solution. Indeed, more concise code that achieves the same effect will generally be preferred.

Basic questions to be answered:

1. Find the empty rows and either fill them with “null” or delete them?

2. Find the highest and lowest EngineSize of each Make and produce the cost difference between them?

3. Find the most expensive Make in both USA and Asia and compare the cost difference with the highest Make in Europe?

4. Which Type has the lowest Horsepower and find their Model?

5. Which DriveTrain is popular in USA?

6. Which Origin has the highest number of Makes?

7. Find the lowest Horsepower with highest EngineSize?

8. List all the Models where their Wheelbase is more than 3000 and their Length is less than 180?

There are more questions to be decided, this is not finalised, just put these as a start to see if anyone is interested.