Track the driving history of people:

I am assuming that the input file is a ".txt" file and that the file is error free in terms of typos. The timing is in the format of a 24 hour clock and that drivers never drive past midnight.

First Step: Parsing through the text while storing the data

Read the information in the file line by line. For each line, split each string and store these strings in an array. This array is then added to another array. Repeat for all lines in the input file.
At the end of this step, we have stored each line in our file as an array within an array.

We return the array of arrays.

An array of arrays data structure is chosen since we just want to be able to access all relevant data for a single entry. Since data is being stored in an array, there exists an order to the information.This order allows us to manipulate the information within the array as per our requirements in the future.

The order within an array is as follows:

1) Command : Driver or Trip
2) Relevant information based on the command: Name of driver or traveling details

Second Step: Formatting the data

In order to be able to store driver's name and the driver's information, it is best to use a hashmap data structure. Since we have a key value pair; key is the drivers name and value is the driver's driving information.

Since we need to store two pieces of information for every driver (average speed and distance travelled) the value in our hashmap will be an array where the first entry is the distance and the second is the average speed.

After deciding how the information is going to be stored, we iterate through the array of arrays and fill in our hashmap appropriately.
Things to consider while filling in the hash map:

1) See information on the same driver twice.
2) Average speed is less than 5 mph or more than 100 mph.

Third Step: Outputting data

The driver with the most distance travelled is returned first. If two drivers have the same distance covered, the driver whose name comes first alphabetically is returned first. Inorder to be able to keep track of the distances and drivers who've driven the same distance,I iterated through the hashmap of drivers and their information to build another hashmap where the key is

the distance driven and the values are an array of drivers who've driven that distance. The list of drivers within our second hashmap was sorted in order of alphabetical precedence.

Finally, to print the data to the console, we store the keys from the second hashmap (where the keys are distances) in an array and then sort this array in descending order. After sorting this array, we use the values of the array as keys for the second hashmap that stores distances as a key and drivers who've driven that distance as values. We use the names of these drivers as keys to access driver information from the first hashmap (where keys are driver names and the values are driver information) and print the appropriate data in the desired order/format.

Testing:

1) Drivers with no trips: works as expected
2) Drivers with same distance travelled: works as expected
3) Different ordering of lines within input data
4) Individual testing of functions within code: modular style of code implemented to be able to trace errors in future to specific functions within the code.