**The below exercises have been broken out into 2 categories: general scripting/logic and mobile automation. Kindly pick 1 exercise from either group & complete in the language of your preference. Check the code into a public github repo and send us the link. The exercise should be in a state such that we can pull from github & execute. Please work independently.**

1. **General scripting / logic**

**Question 1:**

Given a string, reverse each word in it and reverse the entire string. For example, if the given string is “hope you are doing well”, then the output should be “llew gniod era uoy epoh”. Assume that all characters in the string are lower case.

Create a method called reverseEverything(String s) where ‘s’ is the input from the user in order to solve this question.

**Question 2:**

Below is the source code of a sample webpage which displays a food item along with its servings.

<html>

<head>

<body>

<div class=”itemContent”>

<span class="title ng-binding" ng-bind="food.name">Apple butter</span>

<span class="description ng-binding" ng-bind="food.\_servingDesc">1 Tbsp</span>

<span class="title ng-binding" ng-bind="food.name">Apple(s) fresh</span>

<span class="description ng-binding" ng-bind="food.\_servingDesc">1 medium</span>

<span class="title ng-binding" ng-bind="food.name">Applesauce, unsweetened </span>

<span class="description ng-binding" ng-bind="food.\_servingDesc">1/2 cup(s)</span>

<span class="title ng-binding" ng-bind="food.name">Apple(s), dried</span>

<span class="description ng-binding" ng-bind="food.\_servingDesc">1/4 cup(s)</span>

<span class="title ng-binding" ng-bind="food.name">Juice, Cranberry-apple drink</span>

<span class="description ng-binding" ng-bind="food.\_servingDesc">8 fl oz</span>

</div>

</body>

</head>

</html>

Use your knowledge of WebDriver to fetch the following:

1. Display the third and fifth item from the above list. Your output should be:

Applesauce, unsweetened

Juice, Cranberry-apple drink

1. Fetch each food name and its servings and store them in a Map. Iterate through all the entries in the Map and print them.

**Question 3:**

There is a file containing a word and its possible meanings (like a Dictionary). The contents of the file look like this:

Apple – a fruit, a tech firm

Table – an object, contains rows and columns when used in context of computers

Orange – a fruit

Given a path to the file do the following:

1. Create a method called doesFileExist(String path) which takes the path of the file and tells the user if the file exists at that path or not. Assume all paths are relative to your project structure. If the file does not exist, catch the requisite exception.
2. Read each word and its possible meanings and print them out. Your output should look like this:

Word1

Meaning 1

Meaning 2

Word2

Meaning1

Meaning2

Use appropriate data structures wherever necessary.

1. **Mobile Automation Exercises**

**Use an app, device, and language of your preference. Please include any/all binaries, notes, and dependencies.**

**Exercise #1**

Find a financial web application that supplies information on the prices of stocks.

Perform the following actions (for each set of input data):

• Navigate to the application.

• Enter the stock code string for a company (A) exchanged on the NYSE.

• Retrieve the current price of stock A.

• Retrieve the data of the 52 week high and low for this stock over the year from the data provided

on the page (may involve scrolling).

• Output where the current stock price is relative to the high and low end of the range. For

example, “Today’s price of <current> is 10% lower than the 52 week high and 6% higher than the

52 week low.”

• Enter the stock code for a second company (B) traded on the NYSE.

• Compare the Earnings Per share (EPS) of both stocks and output which company has a higher EPS

**Exercise #2**

• Set the location of the device to a North American address (or city center).

• Run the pre-installed Maps application.

• Find a restaurant that is nearby to the location.

• Output the time that Maps reports that it would take to drive to the restaurant.

• Set the location of the device to an address in Europe.

• Calculate the time to drive from this location to another location in the same city in Europe.

• Output the time that Maps reports for the trip.

• Special note:

Select the specific device to run the script by supplying the deviceID to the

capabilities

**Exercise #3**

•  Install a price comparison application on the test device.

•  Select an electronics gadget

• Output the site’s rating of the item.

• Select the option to compare prices for the tablet.

• Output the third lowest price reported for the tablet.