Selectra
HOP! Así de fácil

# Drupal/PHP Code Exercise Programming Instructions

## What you need

To complete this exercise, you will need the following: 1) (Linux) Box with Apache+PHP 7.x installed
2) Your preferred text editor/IDE
3) A web browser
4) Drupal 8.x installed (see requirements for installing Drupal) Once you have installed the above, continue with the instructions below.

## Introduction

You are to choose **one** of the problems below to implement and complete to the best of your ability. This exercise should take a few hours to complete. Please ensure you state any reasons and provide a list of missing functionality if you are not able to provide a complete solution.
You are going to create a web application, create it as a Drupal module.

You are given a tar.gz file containing a basic web application to do all of your work in.

## Setting up your project

1. Unpack the php-code-exercise.tar.gz file into your drupal installation path. Assuming the path to your drupal is /var/www/drupal:

```
$ cp php-code-exercise.tar.gz /var/www/drupal/web/modules/custom
$ cd /var/www/drupal/web/modules/custom
$ tar xvf php-code-exercise.tar.gz
```

This should create the directory selectra.
2. Point your browser to your Drupal administration page and activate the new module
3. You should see a new menu item in your Drupal main menu.

Selectra
HOP! Así de fácil

## What you need to submit

In a zipped up archive, please submit all of the following in one zip or tar.gz file.

- Source code. You should use object oriented programming as part of the solution.
- Unit tests (the more the better)
- Validation against input data
- Demonstrated error handling
- Twitter Bootstrap styles should be integrated into the module. (Without installing the Theme)
- The quality of the design for the result page will be analyzed, mainly its responsiveness
- Comments as deemed necessary
- Documentation (as necessary)
- Any assumptions that you have made

Optionally, you can also include:
- A brief explanation of your solution (such as design considerations) - Class diagrams.

## Problems

You need to choose **one** of the problems below and implement the solution a **Drupal module**.

All problems below require some kind of input. You are to complete this web application that sets up a form, processes the form submission and outputs the results as directed below.

### Problem One: Trains

**Problem**: The local commuter railroad services a number of towns in Kiwiland. Because of monetary concerns, all of the tracks are 'one-way.' That is, a route from Kaitaia to Invercargill does not imply the existence of a route from Invercargill to Kaitaia. In fact, even if both of these routes do happen to exist, they are distinct and are not necessarily the same distance!

The purpose of this problem is to help the railroad provide its customers with information about the routes. In particular, you will compute the distance along a certain route, the number of different routes between two towns, and the shortest route between two towns.

**Input**: A directed graph where a node represents a town and an edge represents a route between two towns. The weighting of the edge represents the distance between the two towns. A given route will never appear more than once, and for a given route, the starting and ending town will not be the same town.

**Output**: For test input 1 through 5, if no such route exists, output 'NO SUCH ROUTE'.
Please use the most direct route and do not make any extra stops! For example, the first problem means to start at city A, then travel directly to city B (a distance of 5), then directly to city C (a distance of 4).

1. The distance of the route A-B-C.
2. The distance of the route A-D.
3. The distance of the route A-D-C.
4. The distance of the route A-E-B-C-D.
5. The distance of the route A-E-D.

**Test Input:**

For the test input, the towns are named using the first few letters of the alphabet from A to D. A route between two towns (A to B) with a distance of 5 is represented as AB5.
Graph: AB5, BC4, CD8, DC8, DE6, AD5, CE2, EB3, AE7

**Expected Output:**
Output #1: 9
Output #2: 5
Output #3: 13
Output #4: 22
Output #5: NO SUCH ROUTE

## Problem Two: Sales Taxes

**Problem**: Basic sales tax is applicable at a rate of 10% on all goods, except books, food, and medical products that are exempt. Import duty is an additional sales tax applicable on all imported goods at a rate of 5%, with no exemptions.

When I purchase items I receive a receipt that lists the name of all the items and their price (including tax), finishing with the total cost of the items, and the total amounts of sales taxes paid. The rounding rules for sales tax are that for a tax rate of n%, a shelf price of p contains (np/100 rounded up to the nearest 0.05) amount of sales tax. Write an application that prints out the receipt details for these shopping baskets...

**Input:**

Input 1: 1 book at 12.49 1 music CD at 14.99 1 chocolate bar at 0.85
Input 2: 1 imported box of chocolates at 10.00 1 imported bottle of perfume at 47.50

**Expected Output:**
Output 1: 1 book: 12.49 1 music CD: 16.49 1 chocolate bar: 0.85 Sales Taxes: 1.50 Total: 29.83
Output 2: 1 imported box of chocolates: 10.50 1 imported bottle of perfume: 54.65 Sales Taxes: 7.65 Total: 65.15

## Problem Three: Mars Rovers

**Problem**: A squad of robotic rovers is to be landed by NASA on a plateau on Mars. This plateau, which is curiously rectangular, must be navigated by the rovers so that their onboard cameras can get a complete view of the surrounding terrain to send back to Earth.

A rover's position and the location is represented by a combination of x and y coordinates and a letter representing one of the four cardinal compass points. The plateau is divided up into a grid to simplify navigation. An example position might be 0, 0, N, which means the rover is in the bottom left corner and facing North.

In order to control a rover, NASA sends a simple string of letters. The possible letters are 'L', 'R' and 'M'. 'L' and 'R' makes the rover spin 90 degrees left or right respectively, without moving from its current spot. 'M' means move forward one grid point, and maintains the same heading.
Assume that the square directly North from (x, y) is (x, y+1).

**Input**: The first line of input is the upper-right coordinates of the plateau, the lower-left coordinates are assumed to be 0,0.

The rest of the input is information pertaining to the rovers that have been deployed.
Each rover has two lines of input. The first line gives the rover's position, and the second line is a series of instructions telling the rover how to explore the plateau. The position is made up of two integers and a letter separated by spaces, corresponding to the x and y coordinates and the rover's orientation.
Each rover will be finished sequentially, which means that the second rover won't start to move until the first one has finished moving.

**Output**: The output for each rover should be its final coordinates and heading.

*Example*:
**Test Input:**
55
12N
LMLMLMLMM
33E
MMRMMRMRRM

**Expected Output**:
13N
51E