This is intended to be a coding exercise done at home, at a time convenient to you, using the tools and libraries that you’re comfortable working with. We will be looking for clean, readable, correct code over code highly optimized for performance. There is no limit to the amount of time you spend on it, but it is feasible to build a complete, working solution in a couple of hours of actual coding time - simpler is better as long as it meets the requirements below!

The focus of this assignment is on implementing logic and algorithms, in particular how to calculate change using a set of available notes and coins. You do not need to provide a nice looking UI (or a UI at all, unless you want to!), it is enough to provide the logic described below through a REST API.

## Problem Constraints

Before looking at the problem itself, please read the following:

* We’re looking for a solution written in TypeScript (preferred) or Python that exposes an API as described below over REST endpoints
* It should go without saying that all code submitted should be created by you for the purpose of this exercise and not taken from other projects or other sources, but feel free to use any general purpose third party libraries
* Please provide the solution in the form of a zip file containing those files you would normally check in to version control, along with instructions for how to run the project
* You don’t need to use a real database or long term persistence of any kind (unless you want to!). Keeping things in memory only is enough for this exercise.

## Building a vending machine

Your challenge is to design and implement a class representing a basic vending machine capable of keeping track of the number of items of each type currently in the machine, the amount of change currently in the machine for each type of coin, and to return correct change given a product selection and a set of coins submitted. You’ll expose ways of interacting with the vending machine through two separate services / interfaces - one mimicking what “regular users” can do, and the other what a maintenance person has access to.

When you create / configure the vending machine, you specify the types of coins accepted and the number of different products sold. The machine then needs to keep track of inventory - the coins in the machine as well as the number of each product available.

You do not need to worry about currencies as such, but the implementation does have to work with coins that are fractions of whatever currency is used, e.g. cents or pennies. For example, the machine should be possible to configure to use dollars, with 5¢, 10¢, 25¢, 50¢ coins supported.

### Operations available to regular users

A user should be able to do the following (operations exposed as REST endpoints)

* Buy a product by specifying the product slot and the amount of coins of different types. The machine should accept a purchase only when a product is available, the user has provided enough money, and it’s possible for the machine to give exact change back to the user. On success, the change is returned (the amount of coins of each type) and the inventory is updated. Note that it’s not enough that the user provides an amount of money equal to or greater than the unit price, the machine has to actually be able to return the exact change for the purchase to go through.

### Operations available for “maintenance users”

These mimic what a maintenance person would be able to do on a physical vending machine, e.g. collect money, restock products, etc. Specifically you should be able to:

* Set the price for a product slot
* Adjust the number of items available for a product slot
* Update the coins available in the machine for each type of coin