Exercise 5: Implementing the `ChocolateMachine` class

Write a class `ChocolateMachine` which represents a machine that dispenses chocolate. Unlike the drinks machine you simulated with the `DrinksMachine` class, the chocolate machine does not have a fixed range of products it delivers. So in the place of the `pressCoke` and `pressFanta` methods, there's a single method `pressButton` which takes a `String` as an argument, where the `String` will be the name of some chocolate item `"Mars"`, `"KitKat"` etc. Instead of returning an object representing the product, the method will return a `boolean` indicating whether a successful purchase has been made or not. In the chocolate machine, each product has its own price.

Here is the complete list of method signatures which should be the public methods of the class:

```
        void insert(int n)
        int getBalance()
        int getPrice(String product)
        boolean empty(String product)
        boolean serves(String product)
        int pressChange()
        boolean pressButton(String product)
        void load(String product,int amount)
        void changePrice(String product,int p)
        int collectCash()
        void add(String product,int price)
        boolean remove(String product)
```

The method `serves` says whether the machine serves a particular product. The method `empty` says whether it has run out of a product it serves. The method `add` adds a particular product to the range of products the machine serves at the price given. The method `remove` removes a particular product from the range served, returning `true` if it has been removed, `false` if it wasn't served previously. The method `getPrice` returns the current price of a particular product. The method `load` adds to the machine's stock of a particular product the number of items given by its `int` parameter. The method `changePrice` changes the price the machine charges for a particular product.

The methods `insert`, `getBalance`, `pressChange` and `collectCash` work as the same named methods worked for the drinks machine example.

You will need to decide on a data structure to represent the working of the machine. One way would be to have an arrayList of objects, where each object in the arrayList represents a product the machine is currently serving, the number of items of that product currently in stock, and the current price of that product.