

## Problem Statement

We have to implement the code for a supermarket checkout that calculates the total price of a number of items.

In a normal supermarket, items are identified using Stock Keeping Units, or SKUs.

To make things simple, we'll use individual letters of the alphabet (A, B, C, and so on) as SKUs in our store for now.

Our goods are priced individually. In addition, some items are multi priced: Buy n of them, and they'll cost you y dollars. For example, item 'A' might cost 50 dollars (Unit Price) individually, but this week we have a special offer: Buy three 'A's and they'll cost you \$130 (Special Price).

Examples of this week's Prices are:

Item	Unit Price	Special Price
A	50	3 for 130
B	30	2 for 45
C	20	5 FOR 80
D	15	

Our checkout accepts items in any order, so that if we scan a B, an A, and another B, we'll recognize the two B's and price them at 45 (for a total price so far of 95).

An Example interface to the checkout would look something like the following:

```
var co = new checkOut (pricingService);
co.scan(item);
co.scan(item);
: :
var OrderTotal = co.CalculateMyOrderTotal();
```

## Expectation

We would require you to implement this checkout which calculates the total price of the checked out items.

We are looking for an object oriented solution for this.

We are looking for automated unit tests to show that we have covered the scenarios.

Please keep in mind that the description doesn't mention the format of the pricing rules. When you design your solution, consider the following so that you will be able to come up with a solution which is extensible.

1. How can these pricing be specified in a way that the checkout doesn't know about particular items and their pricing strategies?
2. How can we make the design flexible enough so that we can add new pricing rules in the future without changing the code?