A simple checkout application

Implement the code for a supermarket checkout that calculates the total price of several items.

In a normal supermarket, items are identified by ‘stock keeping units’ or ‘SKUs’. In our store, we will use individual letters of the alphabet, A, B, C etc, as the SKUs. Our goods are priced individually. In addition, some items have promotions of any of the following types:

- Multi-priced: buy n of them and which will cost you y. For example, item A might cost 50 pence individually but this week we have a special offer where you can buy 3 As for £1.30
- Buy n get 1 free
- Meal deal: buy different items together and get a special price. For instance, items D and E might cost £1.50 and £2 each individually but this week we have a special offer where you can buy one of each for just £3.25

This week’s prices are the following:

Item SKU	**Unit Price (in £)**	**Special Price**
A	0.50	
B	0.60	2 for £1
C	0.25	Buy 3, get one free
D	1.50	Buy D and E for £3.25
E	2.00	Buy D and E for £3.25

Our checkout accepts items in any order so if we scan a B, then an A, then another B, we will recognise the two B’s and apply the special promotion of 2 for £1.

Hint: Because the pricing changes frequently we will need to be able to pass in a set of pricing rules each time we start handling a checkout transaction.

What we are looking for

- Simple design, implemented using either Java or Kotlin (don’t translate Java to Kotlin)
- Test-Driven Development (TDD) and a commit history that shows an incremental approach to tackling the problem (**don’t commit the solution as one single commit**)
- Clean, readable code (don’t use any libraries other than JUnit and optionally Hamcrest for testing)
- Handling of edge cases
- Use of SOLID design principles

What we don’t want you to worry about

- A UI, API, database or reading from files

Submission

Our preference is that you develop it locally keeping the history in Git and then push it to Github or equivalent and send us the link.