Shopping Cart

- Consider a grocery market where items have prices per unit but also volume prices. For example, doughnuts may be $1.25 each or 3 for $3 dollars.
- Here are the products listed by code and the prices to use (there is no sales tax):

Product Code	**Price**
A1	$1.25 each or 3 for $3.00
3-Q	$4.25
45K11	$1.00 or $5 for a six pack
X1	$0.75

- Implement a point-of-sale scanning API that accepts an arbitrary ordering of products (similar to what would happen when actually at a checkout line) then returns the correct total price for an entire shopping cart based on the per unit prices or the volume prices as applicable.
- The program should be an API implemented using Java. You can opt to put a user interface on it or not, but we will only be looking at the API portion. You do not need to provide any form of persistence in this program. Your project should contain some way of running automated tests to prove it works, for example using JUnit.
- The interface at the top level PointOfSaleTerminal service object should look something like this:

```
PointOfSaleTerminal terminal = new PointOfSaleTerminal();
terminal.setPricing(...);
terminal.scan("A1");
terminal.scan("45K11");
... etc.
BigDecimal result = terminal.calculateTotal();
```

- You are free to design/implement the rest of the code however you wish, including how you specify the prices in the system.
- Here are the minimal inputs you should use for your test cases. These test cases must be shown to work in your program:
  - Scan these items in this order and Verify the total price is $13.25:
    A1
    3-Q
    45K11
    X1
    A1
    3-Q
    A1
  - Scan these items in this order and Verify the total price is $6.00:
    45K11
    45K11
    45K11
    45K11
    45K11
    45K11
    45K11
  - Scan these items in this order and Verify the total price is $7.25:
    A1
    3-Q
    45K11
    X1

- Please send us the complete Java project package that we can import in Eclipse IDE to look at your solution. Also, accompany the solution with a concise status document.
- The evaluation criteria are:
    - Understanding of problem statement and ability to meet the requirements
    - Clarity of thought with approach chosen for solution
    - Simple, Clean, Well-factored, OO, Complete Code
    - Complete test coverage
    - Clarity & completeness of communication