

Coding Dojo

[Dojos](#) [Kata](#) [Peoples](#) [Records](#) [Solutions](#)

[Edit on Gitlab](#)

Potter

If you want to try this Kata for yourself or at your dojo meeting, read the problem description and see if the Kata appeals to you. The rest is commentary and helpful clues for if you get stuck solving it. I would recommend trying the Kata for yourself before reading too much of it.

Problem Description

Once upon a time there was a series of 5 books about a very English hero called Harry. (At least when this Kata was invented, there were only 5. Since then they have multiplied) Children all over the world thought he was fantastic, and, of course, so did the publisher. So in a gesture of immense generosity to mankind, (and to increase sales) they set up the following pricing model to take advantage of Harry's magical powers.

One copy of any of the five books costs 8 EUR. If, however, you buy two different books from the series, you get a 5% discount on those two books. If you buy 3 different books, you get a 10% discount. With 4 different books, you get a 20% discount. If you go the whole hog, and buy all 5, you get a huge 25% discount.

Note that if you buy, say, four books, of which 3 are different titles, you get a 10% discount on the 3 that form part of a set, but the fourth book still costs 8 EUR.

Potter mania is sweeping the country and parents of teenagers everywhere are queueing up with shopping baskets overflowing with Potter books. Your mission is to write a piece of code to calculate the price of any conceivable shopping basket, giving as big a discount as possible.

For example, how much does this basket of books cost?

- 2 copies of the first book
- 2 copies of the second book
- 2 copies of the third book
- 1 copy of the fourth book
- 1 copy of the fifth book

answer :

```

    (4 * 8) - 20% [first book, second book, third book, fourth book]
+   (4 * 8) - 20% [first book, second book, third book, fifth book]
= 25.6 * 2
= 51.20

```

Clues

You'll find that this Kata is easy at the start. You can get going with tests for baskets of 0 books, 1 book, 2 identical books, 2 different books... and it is not too difficult to work in small steps and gradually introduce complexity.

However, the twist becomes apparent when you sit down and work out how much you think the sample basket above should cost. It isn't $5 * 8 * 0.75 + 3 * 8 * 0.90$. It is in fact $4 * 8 * 0.8 + 4 * 8 * 0.8$. So the trick with this Kata is not that the acceptance test you've been given is wrong. The trick is that you have to write some code that is intelligent enough to notice that two sets of four books is cheaper than a set of five and a set of three.

You will have to introduce a certain amount of clever optimization algorithm. But not too much! This problem does not require a fully fledged general purpose optimizer. Try to solve just this problem well in order to share it for everyone or even in the ??? . Trust that you can generalize and improve your solution if and when new requirements come along.

- This application has nice application for

Suggested Test Cases

(Originally posted at xp-france)

```

def testBasics
  assert_equal(0, price([]))
  assert_equal(8, price([1]))
  assert_equal(8, price([2]))
  assert_equal(8, price([3]))
  assert_equal(8, price([4]))
  assert_equal(8 * 3, price([1, 1, 1]))
end

def testSimpleDiscounts
  assert_equal(8 * 2 * 0.95, price([0, 1]))
  assert_equal(8 * 3 * 0.9, price([0, 2, 4]))
  assert_equal(8 * 4 * 0.8, price([0, 1, 2, 4]))
  assert_equal(8 * 5 * 0.75, price([0, 1, 2, 3, 4]))
end

def testSeveralDiscounts
  assert_equal(8 + (8 * 2 * 0.95), price([0, 0, 1]))
  assert_equal(2 * (8 * 2 * 0.95), price([0, 0, 1, 1]))
  assert_equal((8 * 4 * 0.8) + (8 * 2 * 0.95), price([0, 0, 1, 2, 2, 3]))
  assert_equal(8 + (8 * 5 * 0.75), price([0, 1, 1, 2, 3, 4]))
end

def testEdgeCases
  assert_equal(2 * (8 * 4 * 0.8), price([0, 0, 1, 1, 2, 2, 3, 4]))
  assert_equal(3 * (8 * 5 * 0.75) + 2 * (8 * 4 * 0.8),
    price([0, 0, 0, 0, 0,
           1, 1, 1, 1, 1,
           2, 2, 2, 2,

```

```
end      3, 3, 3, 3, 3,  
         4, 4, 4, 4]))
```

[Edit](#)[fr](#) [en](#) [pt](#)