```
/* Functional Programming with Scala - Exercise 3
 * Recursion Tree for factoring denominations into money amount
 * Drawn by Richard Walker per Eric Biesterfeld's solution:
 */
 def countChange(money: Int, coins: List[Int]): Int = {
   if (coins.isEmpty) 0
   else if (money < 0) 0
   else if (money == 0) 1
   else countChange(money - coins.head, coins) +
    countChange(money, coins.tail)
 }
```

Change of branch color indicates recursion
Dashed line indicates branching (split) recursion (head series vs. tail series)
Correct outcome/evaluation/stack return: 3