

Infinite Coins

Given an infinite number of quarters (25 cents), dimes (10 cents), nickels (5 cents) and pennies (1 cent), write a method **makeChange()** to calculate and return a Set containing the ways of representing n cents using those coins. Each element of the returned Set should represent the number of coins to compose the entry value, an array like this: [quarters, dimes, nickels, pennies].

The method **makeChange()** can use a Set data structure to store each representation of n cents, and then, return it. A **Set** is a collection that contains no duplicate elements and the order of elements is irrelevant. Consider the following interface defined for Set:

Method signature	Method description
<code>boolean add(Element e)</code>	Adds the specified element to this set if it is not already present (optional operation).
<code>boolean addAll(Set s)</code>	Adds all elements from s that are not already present in this set.
<code>boolean contains(Element e)</code>	Returns true if this set contains the specified element.
<code>boolean equals(Set s)</code>	Compares the specified set s with this set for equality.
<code>Iterator<Element> iterator()</code>	Returns an iterator over the elements in this set.
<code>boolean remove(Element e)</code>	Removes the specified element from this set if it is present (optional operation).
<code>int size()</code>	Returns the number of elements in this set (its cardinality).
<code>Element[] toArray()</code>	Returns an array containing all of the elements in this set.

Table: Set interface

Input example:

n=12

Output for the given example:

[[0,0,0,12], [0,0,1,7], [0,0,2,2], [0,1,0,2]] *

* this is the content of the **Set** which should be returned by the function.

Your proposed solution can be written in **pseudo-code** or any well-known language (C, C++, Java, etc) and you are free to implement any auxiliary functions. Besides, write down a comment to the implemented function, explaining how your function will work like the one below.

```
/**
 * The function below will ...
 * - Obtain the input
 * - Iterate over the elements
 * ...
 * - Print the output and return ...
 */
```