1 - C

2 - C 2 verdadeiro 3 verdadeiro

3- C

4- D

5- A

6 - C

7 - A

8 - nao sei A

```
package a;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

public class Run {

            /*
   quarters = 25;
            dimes = 10;
            nickels = 5;
            pennies = 1;
*/
static int[] coins = {25, 10, 5, 1};

public static void main(String[] args) {

            int numberToTest = 5;
   Set<List<Integer>> result = makeChange(numberToTest);
```

```
    //print every single element inside result
    for (List<Integer> change : result) {
        System.out.println(change);
    }
}

    public static Set<List<Integer>> makeChange(int numberToTest) {

        Set<List<Integer>> result = new HashSet<>();



        List<Integer> currentChange = new ArrayList<>();

        makeChangeEveryCoin(numberToTest, 0, currentChange, result);

        return result;
    }

    private static void makeChangeEveryCoin(int numberToTest, int coinIndex, List<Integer>
currentChange, Set<List<Integer>> result) {

            //if the number is equal 0, there is no point testing
        if (numberToTest == 0) {
            result.add(new ArrayList<>(currentChange));

            return;
        }

            //stop the loop
        if (coinIndex >= coins.length) {
```

```
            return;
        }

        //all number posibilites. Ex: if the max or numberToTest = 25, I can just add once
        int maxCoins = numberToTest / coins[coinIndex];

        //as I said, there is no point testing an infinite range of numbers
        for (int i = 0; i <= maxCoins; i++) {

            currentChange.add(i);
            makeChangeEveryCoin(numberToTest - i * coins[coinIndex], coinIndex + 1,
currentChange, result);
            currentChange.remove(currentChange.size() - 1);
        }
    }


}
```

O que eu mais gosto da faculdade é o tempo extra para dedicar a outras atividades ou outras matérias.

http://github.com/gabrielsantana/prova