Exercise 1. Calculate Change:

**Approach**: When the input is given to calculateChange() method as Long I calculate changes with following approach:

(For input I took Long variable so that I can have more range for numbers (i.e the program won’t give an error for Integer.max\_value) and also when dividing by 25, the remainder is ignored and the result is number of 25 cents coins.

* In calculateChange() method my first if conditions checks how many quarters I can get from given input and if change is possible I will deduct that amount from the input.
* Similarly my second, third and fourth conditions checks how many dimes, nickels and pennies respectively can I get from the remaining input and if changes are possible I deduct that amount from the input.

Runtime complexity: O(1) constant.

**TestCases**: For the test cases I am taking integer values, negative values and a long value to check if my code returns expected values.

**Execute** : Run [EargoCalculateChange.java](https://github.com/DevanshiShah14/EarGo/blob/master/EargoCalculateChange/EargoCalculateChange.java) file on executing it will run my test cases from main method and then it will ask input from the users and to exit the loop press -1.