Technical Assignment - Backend

At Rocket Travel, our team loves travelling. However, many places we travel don't accept our credit cards, and we have to remember to bring dollar bills with us. We've had to correct a few vendors about giving us the correct change from their cash registers: sometimes we get too much and sometimes we get too little. Create a cash register that should be able to accept $20, $10, $5, $2 and $1 bills. If there are sufficient bills of the correct denomination, it should be able to return exact change in each denomination from the cash register. If there are insufficient bills for exact change, it should say so.

Expected Features

Your application should:

- Provide implementation for the following features (see examples below):

`show`	Output the current number of each denomination in the register in format $<total> <# of 20's> <# of 10's> <# of 5's> <# of 2's> <# of 1's>
`put`	Adds some number of each denomination from the register, then print the current state. Same output format as show command.
`take`	Removes some number of each denomination from the register, then print the current state. Same output format as show command.
`change`	Returns change for some amount of money. Output should be denominations of change for the value asked in format <# of 20's> <# of 10's> <# of 5's> <# of 2's> <# of 1's>, e.g. 0 0 4 0 0. This should also deduct the resulting denominations from the register.
`quit`	Exit the program

- Be written in either Java, Kotlin, Groovy or Scala.
- Be readable and logically organized.
- Not dependent on third party libraries, though you can use libraries for testing like junit, spock or mockito.
- Be runnable from the command line.
- Include tests that exercise functionality. One of the test cases we ask people to look for in this code is how to make $8 in change if you have $13 in the cash register.
- Run in jdk 8 or above.
- Include a README

Below is what we expect for a run of a command-line interface implementation of the requirements.

```
/*
 * start program, waiting for command
 */
> java Main ...
ready

/*
 * assume that the cash register was initialized with no bills.
 * put bills in each denomination in: $20's $10's $5's $2's $1's
 * then show the current state
 */
> put 1 2 3 4 5
$68 1 2 3 4 5


/*
 * show the current state of the cash register
 * with the total and each denomination.
 *
 * $Total $20's $10's $5's $2's $1's
 * Total=$68 $20x1 $10x2 $5x3 $2x4 $1x5
 */
> show
$68 1 2 3 4 5

/*
 * put bills in each denomination in: $20's $10's $5's $2's $1's
 * then show the current state
 */
> put 1 2 3 0 5
$128 2 4 6 4 10

/*
 * take bills in each denomination out: $20's $10's $5's $2's $1's
 * then show the current state
 */
> take 1 4 3 0 10
$43 1 0 3 4 0

/*
 * return change for a given amount by showing the number of each denomination the
vendor needs to return: $20's $10's $5's $2's $1's
 * and remove money from the cash register
```

```
 */
> change 11
0 0 1 3 0

/*
 * if there is not enough funds in the register or no change can be made, show an
error
> change 14
sorry

/*
 * exit the program
 */
> quit
Bye
```

Finished Product

Send your completed assignment as a ZIP file or a link to Github, Bitbucket, GitLab or some other public repository. Treat this assignment as though we were going to potentially push this code to production.

The estimated time to complete this task is 3 hours. We respect your time and understand that you have competing priorities. If you find yourself spending significantly more time on this, please let us know, so we can help.

We look forward to seeing what you create!