Problem Statement

GUIDELINES
1) Solution should be in Java
2) It should run on a simple, standard jvm - No GUI, DB or Web Container is required

VENDING MACHINE
The goal of this program is to model a vending machine and the state it must maintain during it's operation. How exactly the actions on the machine are driven is left intentionally vague and is up to the implementor.
The machine works like all vending machines: it takes money then gives you items. The vending machine accepts money in the form of nickels, dimes, quarters, and paper dollars. You must have at least have 3 primary items that cost $0.65, $1.00, and $1.50. The user may hit a "coin return" button to get back the money they've entered so far. If you put more money in than the item's price, you get change back.

Specification:
The valid set of actions on the vending machine are:
• NICKEL(0.05), DIME(0.10), QUARTER(0.25), DOLLAR(1.00) - insert money
• COIN RETURN - returns all inserted money
• GET-A, GET-B, GET-C - select item A ($0.65), B ($1), or C ($1.50)
• SERVICE - a service person opens the machine and sets the available changeand items

The valid set of responses from the vending machine are:
• NICKEL, DIME, QUARTER - return coin
• A, B, C - vend item A, B, or C

The vending machine must track the following state:
• available items - each item has a count, a price, and a selector (A,B,or C)
• available change - # of nickels, dimes, quarters, and dollars available
• currently inserted money

Example input and output
Example 1: Buy B with exact change
Q, Q, Q, Q, GET-B
-> B

Example 2: Start adding change but hit coin return to get change back
Q, Q, COIN-RETURN
-> Q, Q

Example 3: Buy A without exact change (return $.35)
DOLLAR, GET-A
-> A, Q, D