

1h 20m
left

ALL



2

2. Python: Vending Machine

Implement class: *VendingMachine* according to the following requirements:

- can be instantiated using the constructor *VendingMachine*(num_items, item_price) where num_items denotes the number of items in the machine, and item_price denotes the required number of coins to buy a single item.
- has a method *buy*(req_items, money) that represents a buy request where *req_items* denotes the requested number of items, and *money* is the amount the customer puts into the machine. Depending on the state of the machine, one of the following happens
 - If there are enough items in the machine to serve the request and the given money is sufficient to buy the requested number of items, the number of items in the machine is reduced by the requested number of items. The method returns an integer denotes the change given back after the purchase.
 - If there are fewer items in the machine than the requested number, it raises a *ValueError* exception with the message "Not enough items in the machine".
 - If there are enough items in the machine to serve the request but the given amount of money is less than their cost, it raises a *ValueError* exception with the message "Not enough coins".

The class implementation will be tested by a provided code stub and several input files. Each input file contains parameters to test the implementation. First, the provided code stub initializes an instance of the *VendingMachine*. Next, it performs the given operations on the *VendingMachine* instance. The result of their execution will be printed to the standard output by the provided code.

Constraints

- There will be at most 100 operations to be performed

► Input Format Format for Custom Testing

▼ Sample Case 0

Sample Input 0

STDIN	Function
-----	-----
10 2 →	num_items = 10, item_price = 2
4 →	n = 4