Writing Test Cases I
CMPE 287- Spring 2021
Team 04

Rajesh kumar Vartharaj, Rajesh Paturi, Nikhil Prathapani

Points to consider

a. Two types of gumball Red and Yellow. Red worth Nickel, Yellow worth a dime
b. Allow only nickel, dime, quarters. Return any other type of currency
c. Return the balance if any

Program

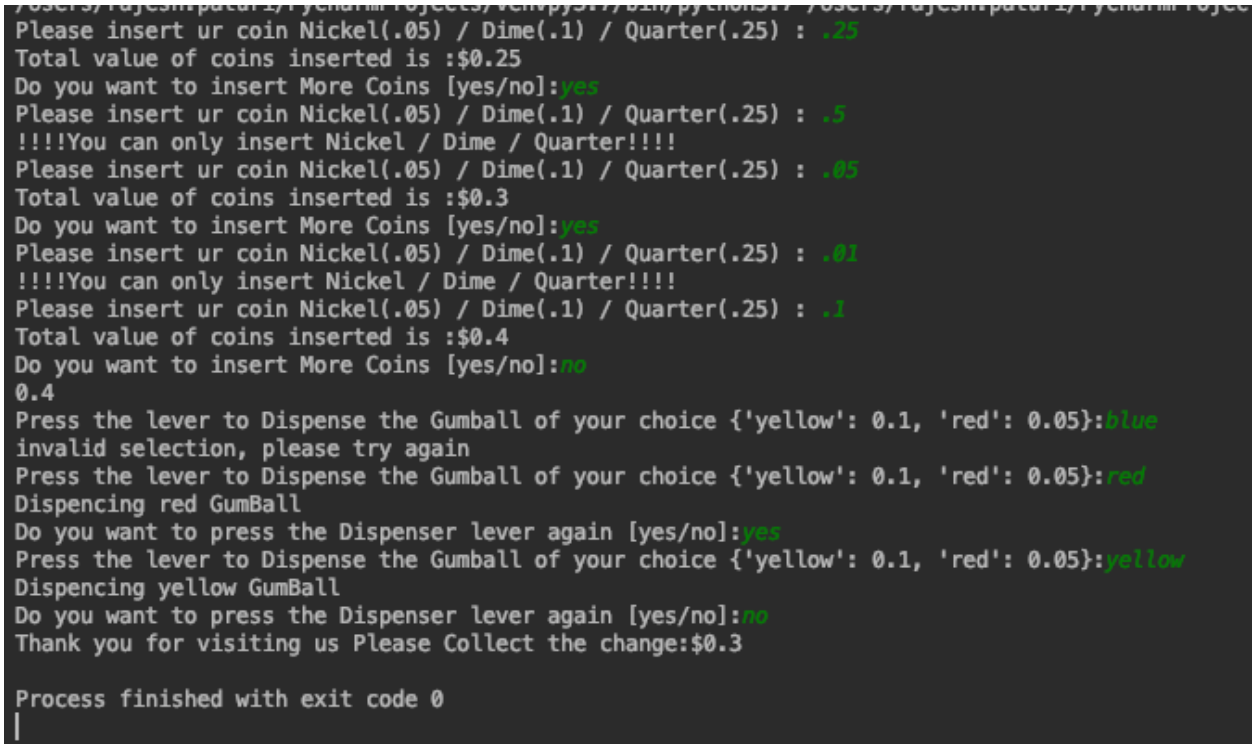
(github link: https://github.com/nikhilp93/cmpe287/blob/main/gumball.py)

```
def collect_coins_from_user():
  total = 0
  acceptable_coin = [0.05, 0.1, 0.25]
  cont = "yes"
  while True:
        coin = float(input("Please insert ur coin Nickel(.05) / Dime(.1) / Quarter(.25) : "))
        if coin in acceptable_coin:
        total = coin + total
        if total > 100:
                print(“Inserted coins exceeds the limit”)
                 return(total)
        print("Total value of coins inserted is :${}".format(total))
        cont = raw_input("Do you want to insert More Coins [yes/no]:")
        else:
        print("!!!!You can only insert Nickel / Dime / Quarter!!!!")
        if cont == "no":
        return(total)
        break
```

```
def dispense_dict(total_paid,gumball_dict):
  gum_ball_colors = gumball_dict.keys()
  cont = "yes"
  rem_balance = total_paid
  while True:
        lever_choice = raw_input("Press the lever to Dispense the Gumball of your
choice {}:".format(gumball_dict))
        #check if the user selected the right lever for dispencing the gumballs
        if lever_choice in gum_ball_colors:
        for i in gum_ball_colors:
        if lever_choice == i and rem_balance >= gumball_dict[i]:
              print("Dispencing {} GumBall".format(i))
              rem_balance = total_paid - gumball_dict[i]
        elif lever_choice == i:
              print("Insufficient Funds to dispense {} GumBall".format(i))
        else:
        print("invalid selection, please try again")
        continue
        cont = raw_input("Do you want to press the Dispenser lever again [yes/no]:")
        if cont == "no":
        return (rem_balance)
        break

if __name__ == "__main__":
  # we have red gumball each worth nickel and yellow gumball each worth dime,
  #Just keep adding to the list,there is no need to modify the code.
  gumball_dict = {'red': 0.05, 'yellow': 0.1}
  #Function to collect coins from the user
  total_paid = collect_coins_from_user()
  print(total_paid)
  #Function to Dispense the gumballs to the end users and return the balance amount
  bal = dispense_dict(total_paid,gumball_dict)
  print("Thank you for visiting us Please Collect the change:${}".format(bal))
```

Sample output from the script



Create test cases to cover all possible scenarios.

Test Cases

S.No	Test Case Name	Steps to Execute	Expected Result
1.	Validating Inserted coins	• Insert Nickel, Dime, Quarter • Insert cent coin	• Customer Inserted coin validated as per the requirement and proceeded to selection. In this case inserted coins are valid • Cent is not a valid coin, it should be returned to the customer and valid error method thrown

2	Validating Customer Selection against coins paid	• Select Red gumball with sufficient balance • Select Red gumball with insufficient coin paid	• Return red gumball and balance coin if available • Return Error message “Insufficient funds” and coins inserted should be returned to the customer
3	Calculate and Verify total paid by Customer	• Insert multiple coins. Example 0.5,0.5,0.1..	• Inserted coins should be added and displayed to the customer. Example: 1.1$
4	Calculate Balance after dispensing	• Insert 0.5 and select Yellow Gumball, Machine should display the remaining balance (0.4) and return it to customer	• Balance 0.4 should be displayed and return to customer
5	Verify remaining balance	• Customer can view balance amount	• All-time customers can view the balance. Example even before selecting the gumball or after selecting the gumball
6	Buy new gumball with the remaining balance	• Insert 0.5, 0.1. Customers can select one red gumball and one yellow gumball OR 6 yellow gumballs. But for every gumball dispense, the remaining balance needs to be calculated and re-used for a new selection. Select one red gumball, and balance should be 0.1, now customer selects yellow gumball	• One Red gumball and one Yellow gumball should be dispensed. Display balance as 0.0
7	(Negative) Overload the gumball machine with unlimited coins	• Insert 101 worth of coins	• Error message should be throw if the coin reached 100 and return total

8	(Negative) Press dispense lever after inserting coin, without selecting gumball	• Insert 0.5 and press dispense lever	• Error message should be displayed to select the gumball type

Test metrics

Possible Inputs/	Red (.05)	yellow(.10)
Nickel(.05)	yes	no
dime(.10)	yes	yes
quarters(.25)	Yes	yes
Balance-Available	yes	yes
Balance-Finished/Low Balance	No	no