preliminary, 2018-10-03

Faculty of Information
School of Graduate Studies
University of Toronto – St. George
Semester: Fall 2018
INF1340H – Programming for Information Systems

Assignment 1 -- 10%
due: 2018 Oct 15, before midnight

Write a Python program that delivers the requested results. Include appropriate documentation.

1. CranJack - a cranberry/jackfruit blend

Spray Ocean has introduced a new drink called CranJack, and it will be sold in MinMax stores in Ontario. Single bottles sell for $1 each, six-packs ("small" packages) sell for $5 each, and two-fours ("large" packages of 24 bottles) sell for $19 each. CranJack is subject to HST -- i.e., it is taxable.

The quantity pricing is not customer-friendly. E.g., six singles do not sell at the six-pack price. Customers buying eight bottles do not get a quantity discount for the two bottles that aren't in a six-pack.

Implement a simple point-of-sale application that will determine how much the customer should pay for the items that he/she brings to the cash register. Then determine how much change should be given back by the cashier for the amount tendered by the customer.

Your program should prompt the cashier to scan the items being purchased, display the total bill amount (amount to be paid), prompt the cashier to enter the amount tendered (what the customer hands the cashier), and display the amount of change to be given back to the customer.

- Pretend to scan items: type in one barcode value per item being purchased.

- Display all currency amounts in d.cc format, always with two decimal places (no more/no fewer). Use the nickel-pricing scheme. I.e., round the bill to the nearest five cents. Assume that all transactions will be settled in cash.

- The "total bill" should display the purchase subtotal before tax, the amount of tax for this purchase, the total (subtotal + HST) before rounding, and the rounded total bill.

- The total bill should be calculated as a running total. As each item is scanned, (re)calculate the customer's total bill, including tax and rounding. In this way, the customer or cashier could decide to stop at any point, and the bill amount would be known then, without further cash register interaction.

- If the customer does not tender enough money, print a friendly error message and try again, or allow the transaction to be cancelled. (You could, for example, treat a tender of 0.00 as a cancellation.)

- For this assignment, it is not necessary to print/display an itemized receipt showing the quantities and prices of every item scanned. The subtotal will suffice.

- At the beginning of each session, display a welcome message that tells the cashier to start scanning. (The customer will be able to see this message, so try to make the wording as pleasant as possible.) At the end of the session, thank the customer for shopping at MinMax.

- Use these barcodes: (universal price codes (UPCs))
  single - 111111
  six-pack - 666666
  two-four - 242424
  done - 0

- Submit your work in a single file unless you have made prior arrangements with the instructor. Identify the starting point by using `if __name__ == "__main__":`

to facilitate marking:

- Use the Quercus assignment dropbox, and if you worked as a team, submit only one copy per pair.
- Provide your name and your partner's name in a comment at the top of the program. Also, provide your section number (L101 for Tuesday, L102 for Thursday).
- Give your program a title (also in a comment at the top), and indicate your revision number and the date of the revision. Your program should be compatible with Python 3.5, and it is suggested that you indicate version compatibility in a comment, too -- i.e., which version(s) of Python will your program run on?
- If you know, at submission time, that your program does not run successfully, please make a note of the known deficiencies in a comment at the top.
- Include appropriate test cases in your docstring.
- Use prescribed names for constants, variables, and functions. (Sorry to take away some of your creativity.) Please draw upon this list as needed:

```
Q_SINGLE = 1          # quantity/size (number of bottles per)
Q_SMALL = 6
Q_LARGE = 24
PRICE_SINGLE
PRICE_SMALL
PRICE_LARGE
UPC_SINGLE
UPC_SMALL
UPC_LARGE
HST_RATE

number_of_singles
number_of_smalls
number_of_larges
subtotal_before_tax
total_before_rounding
total_bill
amount_tendered
amount_of_change

display_welcome()
get_barcode()
calculate_subtotal()
calculate_total_bill()
display_total_bill()
get_amount_tendered()
display_change()
```

example of first lines in your .py file:

```
#
# inf1340, section L101
# assignment 1 - due 2018-10-15
# Lugosi, Bela
# Karloff, Boris
#
# POS program for MinMax
# v1.0.4 - 2018-10-13
# compatible with Python versions 3.4 - 3.7
# source file:  a1-bela-and-boris.py
#
```

scoring: (out of 100 possible)

declaration and use of constants as prescribed	5 marks
headers, docstrings, and bodies of functions:	
to "scan" items	10 marks
to calculate subtotal	15 marks
to calculate total bill, including rounding	15 marks
to enter and consider the amount tendered	15 marks
to display the amount of change	10 marks
other code, including __main__	10 marks
coding style and comments	5 marks
Program produces correct output.	10 marks
facilitation of marking (as described earlier)	5 marks

Don't be late! There is a 15% per day (or part of day) penalty for late submission.