Homework Description

In this homework assignment, you will write a program to simulate a shopping cart for College of Engineering hat sales. The college currently has 2 designs per department (12 designs total) and has labeled them Design 1, Design 2, …, Design 12. Your program will allow the user to repeatedly create new orders (until desiring to quit). During a given order, the program should offer the user options that include (1) print the price and quantity available of each design, (2) add a design to the shopping cart (after verifying that the design has not already "sold out"), (3) remove a selected design from shopping cart, (4) print shopping cart, (5) checkout, or (6) quit. The initial availability of the designs along with their respective prices will be provided in an input file named coeHatDesigns.txt where the first line of the file provides the initial quantity and price (in dollars) of Design 1, the second line provides the initial quantity and price (in dollars) of Design 2, . . . , and the last line of the file provides the initial quantity and price (in dollars) of Design 12. Your program will maintain a shopping cart (an array with quantities ordered by the user) which is initially all zeros. As the user adds/removes designs from the shopping cart, you should make sure that the quantities requested do not exceed those available. After checkout, the inventory should reflect the items have been purchased and the shopping cart should reset to 0 to allow for a new order.

In particular, after reading the input file and appropriately storing the initially available quantities and price for each of the 12 designs, your program should repeatedly present the user with the following options (until the user decides to quit):

(1) **Print the current quantity available and price** for each of the 12 designs to the user. Make sure to present the designs as being numbered from 1 to 12 and *not* from 0 to 11.
(2) **Select available hat design for purchase**:
    - o Ask the user which design she wants to purchase and the quantity to order.
    - o Validate the user input, i.e. keep requesting the Design number until the number is in a valid range (1-12) and make sure that there is enough availability.
    - o If the user selects an item previously selected in the (current) shopping cart, the order quantity for that item should be updated by adding the new quantity to the previous quantity (in other words, if the user requested *3 x Design 9*, and already had *2 x Design 9* in her cart, her cart should now reflect an order of *5 x Design 9*) . Note that order quantities should always be greater than 0. If the user wished to remove an item from the shopping cart, the user would use option 3.
(3) **Remove selected design from purchase**. The purpose of this option is to allow the user to decrease/remove selected designs from the shopping cart. User input should be validated - the user should get an error if she tries to remove *6 x Design 9*, if their cart only contains *5 x Design 9*.
(4) **Print shopping cart**. This should display the current quantities for the shopping cart and compute the total amount that would be due by the user at checkout.
(5) **Checkout**. The program should print the shopping cart, the inventory list should reflect the quantity remaining after the current purchase, and the shopping cart should be reset.
(6) **Quit**. Terminate the program.

HINTS: You need to declare/use several arrays in your program. Here are some suggestions:

- an ***inventory*** array to store the on-hand quantities for the hat designs,
- a ***cost*** array to store the prices the prices, and
- a ***cart*** array to keep track of the user's shopping cart

Arrays should be traversed using a loop. Functions are encouraged to help simplify and organize your program. For example, you could define functions to print the inventory, print the shopping cart, to read the initial inventory from file, to reset the shopping cart, etc – anything that you will (potentially) be doing multiple times. When you define a function that takes an array you should also include an integer argument to pass the length of the array.

Grading Rubric

- 70 points for working code:
  - 10 points for correctly reading the Designs' initial availabilities/prices from a file and storing them in arrays for future use in the program
  - 10 points for correctly displaying list of currently available quantities and prices for each design, as saved in the inventory array.
  - 10 points for correctly adding a hat design to the shopping cart for
  - 5 points for correctly removing a hat design from the shopping cart
  - 10 points for correctly displaying the shopping cart and computing the total amount due
  - 10 points for correctly initializing and resetting the shopping cart when appropriate
  - 10 points for correctly updating the stored item availabilities to reflect the purchase
  - 5 points for correctly having the program repeatedly run until the user wants to quit

  Note that in the items above, the term *correctly* includes performing any necessary error checking

- 30 points for style (see Style Guide on ICON) and readability:
  - style: 5 points for comment block above top of program and above any programmer-defined functions
  - style: 5 points for meaningful variable names
  - style: 5 points for proper indentation
  - style: 5 points for in-line comments
  - readability: 10 points for using arrays/functions/loops as appropriate to avoid excessively redundant code