# Homework Description

In this homework assignment, you will write a program to "model" a traditional vending machine with 12 items potentially available for purchase (e.g., a vending machine with 12 spots to sell items) labeled Item 1, Item 2, ..., Item 12. Your program will allow the user to repeatedly (until desiring to quit): (1) select an item for purchase, and, if the item is still available, (2) pay for the item (using dollars, quarters, dimes, and nickels), and (3) receive change (in quarters, dimes, and nickels). The initial availability of each item along with the cost of each item will be provided in an input file named `vendingMachineItems.txt` where the first line of the file provides the initial availability and cost (in cents) of Item 1, the second line provides the initial availability and cost (in cents) of Item 2, ..., and the last line of the file provides the initial availability and cost (in cents) of Item 12. Item costs can be assumed to be multiples of 5 as the user will only be allowed to pay for items using dollars (100 cents), quarters (25 cents), dimes (10 cents), and nickels (5 cents). In your program, you may also assume that the machine contains unlimited quarters, dimes, and nickels available for change (and that, even for desired change amounts >= 100 cents, no dollars will be provided as change).

In particular, after reading the input file and appropriately storing the initial availabilities and costs for each of the 12 items, your program should repeatedly do the following (until the user decides to quit):

- Present the current availabilities/costs of each of the 12 items to the user and ask the user which item he/she would like to purchase (or if he/she would like to quit). Note: make sure to present the items as being numbered from 1 to 12 and *not* from 0 to 11. In addition, in obtaining the desired item number from the user, your program should keep requesting that an item number be entered until the number is in a valid range.
- (Only if the user doesn't want to quit.) If the item is available:
  - Repeatedly ask the user to "provide" the next dollar/coin until the entered money equals or exceeds the cost of the selected item. Here, you must request each dollar/coin one at a time (e.g., you can't just ask for a total amount to be entered) as the vending machine you are modeling only accepts one dollar/coin at a time. After each dollar/coin is "provided," your program should display the total amount provided so far.
  - Present the change that would be provided to the user (in quarters, dimes, and nickels) where the largest possible number of quarters are provided first, followed by the largest possible number of dimes, followed by the remaining number of nickels needed to provide exact change. (For example, if 80 cents in change should be provided, 3 quarters and 1 nickel should be provided.)

  Otherwise, if the item is not available, a message stating that the item is no longer available should be displayed.

# Grading Rubric

- 70 points for working code:
  - 10 points for correctly reading item initial availabilities/costs from a file and storing them in arrays for future use in the program
  - 10 points for correctly displaying list of current item availabilities and costs as saved in the arrays and requesting that the user select an item (including making sure to display item numbers from 1 to 12 rather than from 0 to 11)
  - 5 points for correctly addressing the case of when the user selects an item number that is out-of-range
  - 5 points for correctly having the program repeatedly run until the user wants to quit
  - 5 points for correctly addressing the case of when the user selects an unavailable item
  - 10 points for correctly updating the stored item availabilities to reflect the purchase of the selected item

  - 15 points for obtaining money from the user correctly (only requesting one dollar/coin at a time until the entered money equals or exceeds the cost of the selected item; ensuring that only dollars, quarters, dimes, nickels may be obtained; correctly keeping track and displaying total amount provided so far after each dollar/coin entered)
  - 10 points for correctly computing/displaying the change that should be provided to the user

- 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