CS360 – Spring 2015
Project 3: Sales Tally Part 2

Using either Java (6/7) or C++ 1998 specification, write a program that solves the following problem.

Input: Sample files posted to Blackboard

1. Obtain manufacturers UPC codes from the ***mcodes.csv*** file by reading the file. There will be exactly one mcodes.csv file in the current directory. The file will be formatted according to the specification below.

    - Column 1 – exactly 6 integers
    - Column 2 – Manufacturers name
    - Columns are comma-delimited
    - Column 2 may be surrounded by two quotes or not surrounded by quotes
    - The file is not guaranteed to be in any sorted order
    - A single manufacturer can have more than one code. This will be denoted by multiple records with different codes but the same manufacturers name in the string. The codes may not be consecutive.

```
000214,"Block Drug Co"
000258,"Thomas J Lipton Co"
000430,"Sandoz Consumer Pharmaceuticals"
742818,"Wizards of the Coast"
081510,"Clairol Inc"
081512,"Clairol Inc"
```

2. Obtain sales data from the sales files that contain a list of UPCs that have sold at a store. The store sales data will be in a file named ***sales.csv*** in the current directory

    - Column 1 - 12 place integer that is the UPC code (the last digit is a check digit)
    - The first 6 values in column 1 are the manufacturers code.
    - Column 2 - <optional> category information (will not be used)
    - Column 3 – Item description
    - Columns are comma-delimited
    - Columns 2 and 3 may be surrounded by quotes. To escape quotes, use double “s (i.e. *978155670488,"9""x12""","Book Native American Stories")*
    - The file is not guaranteed to be in any sorted order

```
742818067003,"15 cards","Magic: the Gathering Revised Edition Booster Pack"
742818083201,,"Wizards of the Coast Pokemon Card Album"
742818083584,,"Pokemon Cardbox"
742818067003,"15 cards","Magic: the Gathering Revised Edition Booster Pack"
742818083034,,"Pokemon Card Holder"
```

```
742818083201,,"Wizards of the Coast Pokemon Card Album"
742818083584,,"Pokemon Cardbox"
742818160193,"10 ct","Magic Unglued Card Pack"
742818160254,"60-card deck","Magic the Gathering: Urza's Legacy: Radiant's Revenge"
742818163026,"36 Booster Pack","Battletech Limited Edition Booster Box"
742818160193,"10 ct","Magic Unglued Card Pack"
742818160254,"60-card deck","Magic the Gathering: Urza's Legacy: Radiant's Revenge"
742818163026,"36 Booster Pack","Battletech Limited Edition Booster Box"
742818165426,,"Magic The Gathering 6th Edition Starter Box"
081510340008,"1 Kt","Frost & Tip Highlighting W/Cap & Hook Dramatic Blonde Highlights"
081512825008,2825,"SUMMER BLONDE LOT OF SUN"
```

Functionality:

New for Part 2: The program should store the manufacturer's codes in a hashtable and a binary tree. A command line switch will determine which is used. The program should default to hashtable. If there is a switch on the command line of –*b* a binary search tree should be used.

Existing functionality: Read in the sales from the store file. You are to provide sales reports by manufacturer and item. You should provide a count for the number of times an item sold (the number of times that item occurs in any store sales file). Organize the manufacturers in alphabetical order. Sales should be reported for a manufacturer only once (combine manufacturers UPCs that are for the same manufacturer). You cannot use a map or any other data structure other than an array. The manufacturers file will only contain codes that are present in the store sales file.

Output sample

```
Clairol
Qty 1 - Frost & Tip Highlighting W/Cap & Hook Dramatic Blonde Highlights
Qty 1 - SUMMER BLONDE LOT OF SUN


Wizards of the Coast
Qty 2 - Magic: the Gathering Revised Edition Booster Pack
Qty 1 - Pokemon Card Holder
Qty 2 - Wizards of the Coast Pokemon Card Album
Qty 2 - Pokemon Cardbox
Qty 2 - Magic Unglued Card Pack
Qty 2 - Magic the Gathering: Urza's Legacy: Radiant's Revenge
Qty 2 - Battletech Limited Edition Booster Box
Qty 1 - Magic The Gathering 6th Edition Starter Box
```

Failure to follow directions will result in point deductions or a zero. There are 85 students in this class. It is unreasonable to expect that any exceptions to the procedure will be made.

Late assignments will not be accepted unless you have a doctor's note covering the entire period from Feb 11 – Feb 22th. Java programs should not require any machine specific code. The C++ programs will be tested on bama.ua.edu. You can gain access to this system for testing. Develop C++ on other platforms at your own risk. Programs must only include constructs made available by the language specifications (Java 6 and C++11/Posix-
https://en.wikipedia.org/wiki/C%2B%2B11) and no other extensions. You also must comply with any prohibitions on data structures in the program description.

Blackboard submission by Feb 22 @11:59pm:
- Program code in ASCII file with .java or .cpp extension. Pdfs of code will not be graded.
- Test results in pdf format. You should generate test data such that it shows that your program works.
- If your solution utilizes more than one source file (and there is good reason why it would), provide a Unix batch (bash or csh), make or ant file to compile and run the test cases. File should run without any machine specific requirements other than make, bash/csh and ant.
- Any relevant notes in text file README

In class printout submission Feb 23 @ 9am:
- Program printout that is easy to read (line wrapping etc).
- Short version of test results (save a tree...)
- The source code (.java or .cpp files) should also be turned in via Blackboard (not emailed to me or the TA). I do not use a Windows computer. Test result submissions of any type other than pdf will not be graded.
- If you forget to turn in your test results, your program may be graded as if it does not work.
- The source code and test results should be printed and brought to class Feb 9th at 9am to get credit. Any assignments that are not submitted via printout on Feb 9th will receive a 50% deduction.
- I will do some initial automated testing. If you name the file something else, I have to go into your code, fix the problem and recompile or rename the file to match your file name.
- This is an individual assignment. The program must represent your own work. You can discuss high-level concepts. Do not show your code to anyone. I reserve the right to ask you about your program to discern if you indeed wrote the code. If you cannot explain your code and choices verbally, you may be turned in for academic misconduct. All submissions will be analyzed to identify possible cases of cheating. Any cases of suspected collaboration will be referred to the College of Engineering Dean. A zero or low grade is always better than having an academic misconduct on your academic record.

Here is an overview of evaluation criteria. It is subject to change within 10% at my discretion. The top submissions that contain particularly elegant and clear code may earn extra credit.

70% - Functionality

- **Must use student-written hashtable and binary search tree for storing manufacturers codes/descriptions for lookup**

Program displays a reasonably fast method for producing the output

- Program contains the correct code to produce the output.
- Only read the input files once
- Program meets the requirements (correct calculations and categorizations, data structure, etc)
- Only hardcoding includes the data file name
- No artificial limits on the number of manufacturers or items (of course computation will limit us)
- Output from running the program is included that shows that the program works
- Program uses appropriate language constructs
- Overall design should be reasonably consistent with OO design principles

30% - Design

- Program exhibits defensible design choices in algorithms and data structures (if you add any)
- Program does not contain extra loops or any code that hurts efficiency
- Program must use appropriate and consistent style for naming of elements
- Program must include reasonable whitespace and appropriate indentation
- Program must include comments, especially in areas where you need to support your choices or where the purpose of the code is unclear.

** Clarifications on the assignment will be posted to blackboard. You will be responsible for consulting blackboard frequently and remaining aware of clarifications.