Examine the file shopping\_data.txt. It contains the transaction number followed by a list of items that were purchases. Implement the Apriori Algorithm to process the file. Your job is to find all item sets with minimum support of 1%. Here is the expected output:

{1=[[0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49]], 2=[[0, 2], [0, 46], [1, 19], [2, 46], [3, 18], [3, 35], [4, 9], [5, 14], [5, 22], [5, 31], [5, 42], [7, 11], [7, 15], [7, 37], [7, 45], [7, 49], [9, 13], [9, 45], [11, 37], [11, 45], [12, 31], [12, 36], [12, 48], [14, 22], [14, 44], [15, 49], [16, 32], [16, 45], [17, 29], [17, 47], [18, 35], [23, 24], [23, 33], [23, 40], [23, 41], [23, 42], [23, 43], [24, 40], [24, 41], [24, 43], [27, 28], [29, 47], [31, 36], [31, 48], [32, 45], [33, 42], [36, 48], [37, 45], [40, 41], [40, 43], [41, 43]], 3=[[0, 2, 46], [3, 18, 35], [7, 11, 37], [7, 11, 45], [7, 15, 49], [7, 37, 45], [11, 37, 45], [12, 31, 36], [12, 31, 48], [12, 36, 48], [16, 32, 45], [17, 29, 47], [23, 24, 40], [23, 24, 41], [23, 24, 43], [23, 40, 41], [23, 40, 43], [23, 41, 43], [24, 40, 41], [24, 40, 43], [24, 41, 43], [31, 36, 48], [40, 41, 43]], 4=[[7, 11, 37, 45], [12, 31, 36, 48], [23, 24, 40, 41], [23, 24, 40, 43], [23, 24, 41, 43], [23, 40, 41, 43], [24, 40, 41, 43]], 5=[[23, 24, 40, 41, 43]]}

Design considerations. These are very vague on purpose, feel free to change them.

Create an ItemSet class that stores an ArrayList<Integer> (the items). Add appropriate methods.

In your Lab5.java class, store the following variables.

ArrayList<ItemSet> transactions //lists of all itemsets

ArrayList<Integer> items //lists of all items

HashMap<Integer, ArrayList<ItemSet>> frequentItemSet //lists frequent imtemsets. E.g., for key=1, store all 1-itemsets, for key=2, all 2-itemsets and so on.

In the Lab5.java file, create the following methods, and possible some other.

public static void process(String fileName) //processes the input file  
public static boolean findFrequentItemSets(int k) //finds all k-itemsets, Returns false if no itemsets were found (precondition k>=2)  
public static boolean isFrequent(ItemSet itemSet) //tells if the itemset is frequent, i.e., meets the minimum support  
public static void findFrequentSingleItemSets() //finds all 1-itemsets  
main //the main method