The purpose of this assignment is to gain familiarity with dictionaries.
A homophone is one of two or more words that are pronounced alike but are different in meaning or spelling; for example, the words “two”, “too”, and “to”.
Write a Java program that uses one of the implementations of the dictionary interface that we have discussed to find the largest set of words that are homophones.
Use only UALDictionary, OALDictionary or BST, and do not use data structures that we have not covered so far in the course.
The file “cmudict.0.7a.txt” on Vocareum contains a pronunciation dictionary downloaded from
http://www.speech.cs.cmu.edu/cgi-bin/cmudict
The page also contains a detailed description of the pronunciation dictionary. The file consists of lines of the form
ABUNDANT AH0 B AH1 N D AH0 N T
The first string is the word, which is followed by one or more phonemes (or phones) that describe the pronunciation of the word. There are 39 phonemes occurring in North American English that are used in the dictionary. The collection of 39 symbols is known as the Arpabet, for the Advanced Research Projects Agency (ARPA), which developed it in the 1970’s in connection with research on speech understanding.
The starter code on Vocareum contains several files you can use. UALDictionary.java is a class that implements the dictionary interface using an unordered array list. OALDictionary.java is a class that implements the dictionary interface using an ordered array list. The array list stores (key, value) pairs as described in the class KVpair.java. BST.java is the dictionary implementation based on a binary search tree (associated classes are also in the starter code). Pronunciation.java is a class that stores and manages access to a (word, phonemes) pair.
There is also a program Homophones.java that shows how you can read in the cmu dictionary (skipping comments and so on). You can modify that program so that when you read in a pronunciation entry you store it in an appropriate dictionary.
Call your program MaxHomophones. The input is a single positive integer n. If the largest set of homophones in the first n lines of the pronunciation dictionary is of size k, then your program should print out k on the fist line, followed by k homophones, each on a new line. If there is more than one collection of k homophones, your program should print out each group, separated by a blank line. The output should be all upper case for consistency with the pronunciation dictionary.
For example, if the input is
1000
then the correct output is
5
ABBE
ABBEY
1
2
ABBIE

ABBY
ABIE
If the input is
40000
then a correct output is
10
BURY
BUERRY
BERRY
BERRIE
BERRI
BERREY
BARRY(1)
BARRIE(1)
BARRE
BAREY
BAYLY
BAYLEY
BAYLEE
BALLY(1)
BALEY
BAILY
BAILLY
BAILLIE
BAILIE
BAILEY
It would also be correct if the order of the two groups of homophones was reversed.
If the input exceeds the size of the dictionary, the output should be the same as if the input were the size of the dictionary.