CS 251
Fall 2007
Voicu Popescu
Assignment 4
Due Monday September 24, 11:59PM

Vectors, Lists, and Sequences

1. Implement a C++ class that models a collection of words with an array of lists. Words are spelled exclusively with lower case letters. There is a list for each array element. Each array element corresponds to a different lower case letter. A word is stored in a list according to its first letter. The word collection can store duplicates. The data structure should not make any assumption about the maximum number of words. The maximum length of a word is 20 letters. The word letters are stored in a null terminated character string (i.e. there is a 0 after the last letter of the word; a 0 and not a '0'). Your data structure should provide:
    a. A **constructor** that makes an empty word set (array elements have empty lists).
    b. An **insert** method that adds a given word to the word set in constant time.
    c. An **isPresent** method that output "Yes" to standard output if and only if a given word is already part of the word collection; running time $O(n_l)$ where n_l is the length of the longest list.
    d. A **remove** method that removes one occurrence of a given word, if any; running time $O(n_l)$, as before.
    e. A **removeAll** method that removes all occurrences of a given word, if any; running time $O(n_l)$, as before.
    f. A **removeDuplicates** method that eliminates all but one occurrences of all words in a collection.
    g. A **print** method that takes a character input, and input all word start with that character.
    h. A **printAll** method that output all lists, alphabetically.

2. You notice that many words begin with the letter *'p'*, which makes the list for *p* long and the isPresent method inefficient. Describe a modification to your data structure that alleviates this problem, and give a pseudocode description of the insert and isPresent methods for the modified data structure.

3. Extra-credit: Problem C-5.6, page 247 of text book. (3%)

4. Turn in instructions:
    a. Assignment specific Assignment specific
        i. For question 1, you must use the main file and empty class files provided (see code.zip). The main file constructs a word collection object, and then parses <<command> <data>> lines from the standard input and calls the appropriate class methods. You CAN NOT use the Standard Template Library (STL) or any other libraries, you must implement the data structure yourself. Include any additional classes (e.g. list class) in the WordCollection.{cpp, h} files.

    ii. For question 2, turn in a pdf (not ps) document named a4.pdf, place it in the lab4 directory.
    iii. For extra credit question 3, turn in a pdf (not ps) document named a4extra.pdf, place it in the lab4 directory. Remember to turn in the blank extra credit file.
    iv. Include a Makefile and have it build the programs as follows by having separate rules: (the executable must be main (lowercase) or it won't be graded. Missing clean rule will also cause deduction)
        make – creates the executable main
        make clean – deletes all object and executable files
    v. Use lab4 for labX and all your files MUST be in a directory named lab4 and in no other subdirectory.
    vi. The command will be /turnin -c cs251 -p lab4 lab4 where all your files are in the lab4 directory and you are in the directory above lab4

b. General:
See instruction at http://www.cs.purdue.edu/cgvlab/courses/251/As/turnin.html
Make sure to check it EVERY TIME before submitting a new assignment. It is kept updating.