

Programming Assignment 3 (draft)  
Due 11:59 pm, Monday, October 26  
40 points

To get credit for this assignment, create a subdirectory named prog3 and do all of your work for this assignment in that directory. The required files will be electronically collected at the specified due date/time. You should copy the file p3.txt from the class library to this directory.

Objectives.

- Practice program/class design
- Using the STL list class
- Using iterators
- File Processing

Write a program that will read a text file and create a *line index* of each valid word in the file. The words in the index must be in alphabetical order with no duplicates. The list of line numbers for each word must be in ascending order with no duplicates.

The program should read from the file “p3.txt” and create the output file “p3index.txt”. For the purpose of this assignment, a valid word is any sequence of characters from [‘a’..’z’, ‘A’..’Z’]. All words must be converted to lower case before indexing.

Credit Levels:

**C-level: 32 points.**

You must use the STL string and list classes in a meaningful way.  
Hint: use sort and or unique list class functions.

**B-level: 36 points possible C-level +**

New line numbers must be inserted into the list in the correct position; i.e. the list of line numbers for each word is maintained/created in sorted form. (this means you may NOT use the built-in sort and unique functions to maintain line number ordering)

**A-level: 40 points possible B-level +**

New words must be inserted into the list in the correct position; i.e. the index is maintained/created *in sorted form*. (this means you may NOT use the built-in sort and unique functions to maintain index ordering)

Your program should also determine and print

- the word(s) that occur(s) on the most number of lines (for the purposes of this calculation, a word occurring more than once in a line counts as one occurrence)
- the number of words that occur least frequently.

Required files:

p3a.cpp or p3b.cpp or p3c.cpp // main application file  
readme.txt // containing:  
    list of all files needed for this application  
    compiling instructions  
    A paragraph describing your solution approach  
class header files  
class implementation files.  
p3.photo containing (in order):  
    cat readme.txt  
    g++ compile line  
    a.out  
    cat p3index.txt  
    cat p3alblc.cpp  
    cat header/implementation files

Output format:

<word>:<comma separated list of line numbers>  
(use field width of 16 for the word)

Example:

```
1234567890123456
           and:1,2,3,19
disambiguate:2
           hello:2,5,6,7
```