COP3330 Programming Assignment 8

A variation of the UNIX wc utility

Objectives

* Practice problem solving using C++

Project details

Your program should read the standard input until it reaches the end, counting the number of times each word/character is used. A word can either be an identifier or a number. An identifier is defined as a letter followed by a sequence of letters or digits('a'..'z', 'A'..'Z', or '0'..'9'). Identifiers are case insensitive ("AA00", "Aa00", "aA00", and "aa00" are the same). A number is defined as a sequence of digits ('0'..'9') that are not in an identifier. Different sequences represent different numbers. For example, number "001" is different from number "1". Identifiers are separated by non-letter and non-digit characters. Numbers are separated by identifiers or non-letter and non-digit characters. Your program should record the number of times each identifier/number/character happens. It should first output the number of lines, words, and characters. After that, it should output the five most used characters, the five most used numbers, and the five most used identifiers as well as the number of times these characters/numbers/identifiers are used. Since identifiers are case insensitive, the program only outputs identifiers with lower case letters. The characters, numbers and identifiers should be outputted in the descending order based on the number of times they are used. When two characters happen the same number of times, the character with a smaller ASCII value should be considered as being used more frequently. When two identifiers (numbers) happen the same number of times, the identifier (number) that occurs earlier in the input should be considered as being used more frequently. An example executable (for the program machines) 'proj8\_linprog' is given. You should make the outputs of your program the same as those of 'proj8\_linprog'. When printing characters, use '\t' for tab and '\n' for newline. All other letters including invisible ones should be outputted normally.

You can assume that the total number of different words used in a file is less than 30000 and the number of characters in a word is less than 100.

Special Note:

If you did not score 70 or more in programming assignments 1 or 4, YOU MUST score 70 or more in this assignment in order to get a C- or better grade for this course. This is a ABET/SMALC requirement for passing this course.

Submission

The due time for this assignment is July 17 (Wendesday), 2013. 11:59pm.

Put all of the implementation in one file named proj8.cpp. Tar proj8.cpp and name the tar file yourlastname\_firstinitial\_proj8.tar and submit the tar file in blackboard.

Grading policy

The program must work on linprog. O point for programs with any g++ compiler error on linprog.

* ‘g++ proj8.cpp’ has no compiler error (20 points)
* Tar file name, comments in program, etc (5 points)
* Program counts the number of characters correctly (30 points)
* Program counts the number of lines correctly (15 points)
* Program counts the number of words correctly (10 points)
* Program outputs the most used letters list correctly (10 points)
* Program outputs the most used words list correctly (10 points)

Hints

1. Start the project as soon as possible. This program is larger than other problem solving programs that you did before.
2. You should start by counting the number of characters and the number of lines in the input. This will get you 70 points for the assignment with a very small program.

SMALCs/ABET assessment rubric:

1. effective use of storage facilities (variables, objects, arrays) in program solution
2. effective use of functions in program solution
3. effective use of control structures (conditionals, loops) to create algorithmic solutions for the specified program features
4. effective performance of input/output
5. Code compiles and is readable.
6. Overall ability demonstrated in translating the program specification

to an appropriate solution