CS280 Programming Assignment 1
Fall 2018

Finding matching strings is a common problem in computing. For this assignment, we want to look at WORDS and PHRASES in our input. A WORD is a sequence of characters, separated by whitespace (that is, spaces, tabs, newlines). A PHRASE is a sequence of WORDS separated by a space between the words. The length of a PHRASE is the number of WORDS in the phrase.

When processing input, all letters should be converted to lower case.

This means that an input file that looks like this:

> This isn't the pr3oper time, YOU know, "DuDe", to fight about this!

And is treated as these words:

```
this isn't the pr3oper time, you know, "dude" to fight about this!
```

When we run our program, we consider phrases of a certain length N -- that is, all sequences of N consecutive words. Using the example above, here is the list of all of the phrases in the input file for the case of N equal to 6:

> this isn't the pr3oper time, you
> isn't the pr3oper time, you know,
> the pr3oper time, you know, "dude"
> pr3oper time, you know, "dude" to
> time, you know, "dude" to fight
> you know, "dude" to fight about
> know, "dude" to fight about this!

The assignment is to write a program that finds the most common expressions of length N across all input files. N will be the first command line argument, then a mode ("all" or "top"), followed by the names of all of the input files.

If the mode is "all", print all phrases, whether the top frequency or not. If the mode is "top", then only the ones that are max frequency. In either mode, results are printed in alphabetical order by the phrase text.

Your program must process the command line arguments. Then, for each input file, your program shall:

- Read each file, according to the rules described above, resulting in a list of WORDS
- Determine PHRASES of N words
- Find and print the PHRASE that appears the most often by examining the map element(s) that has the highest frequency (or all of them if in "all" mode)

After processing every file, your program should print all common phrases across all files. This means, for example, that if every file contains the phrase “game over man” you must recognize that and print out that phrase.

The format of your output for EACH file should be:

    PHRASE (# of times it appears)

Where FILENAME is the name of the file, and PHRASE is the phrase.

Note that it’s possible that there are multiple phrases that appear with this highest frequency. For example, with N == 2, the input:

    This is mine. This is yours.
    These are his. These are hers.

Contains the phrase “this is” 2 times and “these are” 2 times. For that situation, the phrases would be printed out as follows:

    this is 2
    these are 2

As you work on this problem, take note of the following possible error cases:

- You must detect the case where no arguments are passed. In that case, print the error message “NO PHRASE LENGTH” and stop.
- You must detect the case where the argument specified for N is not an integer. In that case, print the error message “INVALID PHRASE LENGTH” and stop, also if the integer is negative
- Print “NO MODE” if a second argument is missing
- Print “INVALID MODE” if the second argument is something other than “all” or “top”
- You must detect the case where there are no file names specified. In that case, print the message “NO FILES GIVEN” and stop.
- If a file cannot be opened or read for any reason, print the error message “BAD FILE FILENAME” and continue to the next file.
- If there are no phrases at all, i.e. all empty files or no phrases long enough for the size given, then it should print “NO PHRASES”
- Note that it is also possible that ALL of the phrases in the input occur with the same frequency. In that case, there is no phrase that appears more often than any other. In that case, the output should be “ALL PHRASES EQUALLY FREQUENT”

The program will be submitted and graded in separate parts:

Part 1

All of the error cases other than “ALL PHRASES EQUALLY FREQUENT”, including opening files, but NO file processing (i.e. just open and close the file to see if it’s there - no reading, no parsing, no counting, no printing)

Part 2

Processing all files and printing “all”-mode phrases with frequencies, but only for phrase length 1 (i.e. one word)

Part 3

Processing all files, including printing either the top or all mode, for phrase lengths that might be greater than 1, and determining if all phrases are the same length

Note that both in your “work” directory, and also on Moodle, there are files:

- cases.txt
- cases.tar.gz

The cases.txt file covers all the possible cases, as in what arguments are used for each test case.

The cases.tar.gz is a zipped file containing all the input and output files.