

Daniel Domalik
CIS3190
Michael Wirth
04/6/2018

Assignment 4, Flesch Readability Index

To start, I will run through a description of how the program operates. When you run the program, it will display a short introduction as well as some rules. The user is then prompted for his/her input, a file name. The program then scans through the file and grabs the number of sentences, (most of) the syllables, and (most of) the words. The program then uses the formula to calculate the Flesch readability index and a function is then called to calculate the Flesch-Kincaid grade level.

A simpler version can be displayed like this:

1. Display rules
2. Prompt user for input
3. Read in file contents line at a time
4. Collect sentence number
5. Collect syllable number
6. Collect word number
7. Loop until end-of-file

The process of converting this ancient language to Ada was quite troublesome. The program is not complete. It does not compute the proper number of words and syllables but is close. The way I went about converting the program was first drawing out a flow chart of how the program operates. Then I broke each function or check down into each of their algorithms. I then attempted to replicate that in my program. One thing I should have stayed true to was instead of reading in the whole line from the file, I should have separated the words into a string array of some sort. This way, it would be more true towards the original program and instead of keeping track of where you are in the huge string. Another advantage of converting everything to a string array is that it would be much much easier to count everything. If the last character of a value in the string array was a sentence closer, you would just add one to that counter. If I had more time and did not start this program late I would have done that. Instead I threw together a somewhat jumbled interpretation of the original program. Perhaps it does work and I am being too hard on myself. To test this I compared the case at the end of the documentation. The results I had gotten were different, however, the author himself says that although there are exceptions to the rules, it works in most number of cases.

Understanding the original program syntax was not an issue. It was quite English-like, similar to what I wrote about COBOL: "I enjoyed COBOL because of how human-like it was. Personally, it seems like a programmer would use the English language to try to write code. Things like DISPLAY and OPEN INPUT are very English/Human-like. This being compared to something like C where printing something to the screen takes up all the space on a line in your IDE/text editor." Especially how in the PDF provided, there was documentation about how the program operates and the algorithms that help it function. Especially this:

“In general the text analysis portion of the program uses the following rules:

1. Periods, exclamation points, question marks, colons and semi-colons count as end-of-sentence marks.
2. Each group of continuous non-blank characters counts as a word.
3. Each vowel (a, e, i, o, u, y) in a word counts as one syllable subject to the following sub-rules:
 - 3.1. Ignore final -ES, -ED, -E (except for -LE)
 - 3.2. Words of three letters or less count as one syllable
 - 3.3. Consecutive vowels count as one syllable.

Although there are many exceptions to these rules, it works in a remarkable number of cases."

All in all, this program was quite fun. I like a somewhat challenging algorithm puzzle. The conversion from the older language was not a problem, I understood it just like COBOL. Saying good things about PL/I, however, does not mean I would like to pursue a career in it. The variable declaration, just like COBOL, is too tedious for my liking.

P.S.

I'd just like to thank the professor, Michael Wirth, for an awesome course. Sadly, there aren't many intriguing electives to take in the CIS field at the University of Guelph. This course, however, was enjoyable and gives great insight into how much (and how little) languages have evolved through the years.