CIS 3190
Flesch Index Calculator Reflection Report

James Singaram
0758172

April 5, 2021

One of the early exercises I was given when learning programming was to create a flesch index calculator in C and analyze a number of famous historical works. Being given that task again presented an opportunity to reflect on the progress I have made and examine the differences in my coding style that have developed as a consequence of expanding into additional languages.

The very first step in the design process was to refamiliarize myself with the flesch index and how it is derived. While the provided documentation gave a brief overview of the steps involved in counting the number of syllables, words, and sentences, reviewing guidelines and procedures written by other authors provided a deeper understanding on the topic.

Before any work could be done on the Fortran version of the program it was necessary to create a rough outline that it would follow. Despite having no previous experience with the language, an initial skim of the PL/I code was fast and the major steps taken by the program were easy to identify. After that the PL/I code and the code for my C flesch index calculator were read in parallel and both of their actions catalogued. These two lists were then combined to create a flowchart that the Fortran program would follow. The main emphasis of this step was to understand the broad actions taken by the programs more than the exact way in which a step was accomplished. Language-specific implementations were less important than knowing that a step had been taken.

One major change from the PL/I code that I wished to implement was to have the calculator read the entire text file into a single, dynamically allocated string. The PL/I version read the file line by line with a maximum line length of 80 characters, however that seemed to be an unrealistic assumption for modern documents. A cursory search suggested that Fortran 2003's deferred-length character strings may have been suitable for the task, so it was decided to upgrade the new program from Fortran 95. This was successful in reading lines of varying lengths, however challenges arose in reading the entire document at once and the idea was scrapped. Despite deciding not to use deferred-length character strings, significant time had been spent getting to that point and the potential errors that could be introduced by reverting to an older fortran version were seen as a greater liability than continuing with Fortran 2003.

The process of counting syllables, words, and sentences was adjusted to more closely resemble the C code than the PL/I code. This was primarily done to enhance readability. While the PL/I code is very dense and requires far fewer lines, it is not as immediately apparent how it functions. Making the Fortran code easier to review and modify was not necessary during this project, however it would have been worthwhile if any major bugs had been present or if the program was to be updated in the future. Unlike the PL/I version, colons and semicolons were not included as sentence terminators.

Overall this project followed a similar design process as previous assignments. Being entirely unfamiliar with the source language complicated matters, though this was mitigated by reviewing how the tasks were accomplished in other languages and comparing the results. After a rough flowchart was produced it was a simple matter to translate it into Fortran. Investigating the features of different Fortran versions and having to select a version to use was an unanticipated hurdle, and the lessons learned from it seemed an appropriate capstone for the course.