

CIS*3190 Assignment 1 Reflection Report

Over the last couple weeks, I enjoyed learning Fortran. With the programming experience I already have, I thought that the language was pretty easy to learn as it seemed similar to C and Python due to it being a high-level language. I found keeping track of arrays and formatting input/output to be pretty similar to C, while the ease of performing calculations and the many built-in functions was like Python. Due to the similarity to these languages which I'm familiar with, I found it wasn't too difficult to complete this assignment.

Personally, I think there are a couple reasons that make Fortran a good language. The first thing I found very convenient was a large list of built-in functions that don't require you to import any extra libraries. This helped me a lot in the assignment when I was casting variables to another type or performing math operations like modulus. Fortran is also very fast and efficient when doing calculations using matrices, which for computational programming is very useful when working with huge data and calculations. I also found using modules to be very simple and intuitive, as opposed to in C where you must make libraries and header files, which I struggled with greatly when I first started programming larger programs in C. One final reason I find that Fortran is a good language is that it is strongly typed. I find that debugging strongly typed languages is a much easier task, as the compiler will let you know if you are using a variable incorrectly. This greatly helped me while I was programming the second task of this assignment, as when I was converting characters to integers and vice versa I found myself doing it incorrectly. However once I tried to compile, the compiler let me know immediately where my mistake was which allowed me to make a quick fix. If something like that happened in a loosely typed language like JavaScript it would've taken me much more time to fix it.

Based on many of the reasonings above, I think using Fortran for this program was easier than it would've been in C. Especially since the lucifer algorithm does lots of computations for matrices, doing this program in Fortran is much more efficient. Another reason Fortran is better than C for this program is the use of multi-dimensional arrays. This is something that would be very difficult to do in C. Keeping track of your allocated memory and where your pointers point would make this program a lot more difficult, especially with 3D arrays where you would have to use triple pointers. Using C for this program would also cause issues when you need a dynamically sized array in the `expand()` and `compress()` functions. Having to use `malloc` and `realloc` would be much harder than just simply using `dimension(0,*)`. Overall, I think Fortran is a great language for this program as it uses most of the language's strengths.

While doing the re-engineering, the greatest problem I faced was restructuring the implicit variables as well as documenting the code. At first I wasn't sure exactly how the algorithm worked, so I wasn't sure what type to cast the implicit variables to. This also made it difficult for me to be able to document the code, as I found the algorithm very complicated and did not understand what the variable names represented. However I found that after re-factoring the code, it seemed a lot cleaner and easier to understand. Fortran 95 is a lot more similar to languages I've used before than Fortran 70, and I found that the formatting changes make the program flow more intuitive which in turn made it easier for me to understand.