

Assignment 1 Reflection Report

For the first task of this assignment we had to re-engineer the original lucifer program in FORTRAN 77 to run in Fortran 95. This was quite challenging for many reasons but the main problem I faced was understanding what the algorithm was doing. It was very difficult to follow the original program due to the lack of readability. I found re-writing the do loops took the most time due to the lack of indentation. It was hard to follow where each loop ended in multiple nested loops. I also found it difficult to understand what the equivalence statements were doing since I had not previously seen this type of statement in other languages. Because of this I spent a very long time trying to take out the equivalence statements to use the newer reshape function instead.

Although Fortran is a legacy language it still contains many features that give it an advantage over languages such as C. One of these features is the way Fortran handles arrays. Arrays in Fortran are much easier to use than arrays in C for a few reasons. One is that the user does not need to worry about memory allocation. This makes the creation and handling of arrays and strings clearer. In Fortran the user is also able to create an array of any index range. Fortran also allows the reshaping of arrays which is much more efficient for data handling than having to duplicate the memory into a new array. Another way that Fortran is more efficient than languages such as C is that variables are passed by reference instead of by value. This makes the transfer of data between subroutines much faster and lessens the risk of losing localized data.

After considering the differences between Fortran and C I don't think this program would have been easier to write in C. As mentioned above, the ability to reshape arrays is vital to the lucifer algorithm and the easy use of array indexing without worrying about memory allocation allows Fortran to be much more efficient for numerical and mathematical programs. The lucifer algorithm relies on numerical computation and is therefore a good candidate for Fortran as Fortran was designed to handle such programs. Not only was the program easier to write in Fortran but it is also much more efficient.

Although I had some difficulty re-engineering the original algorithm code, I found that Fortran wasn't too hard to learn. I found that my previous knowledge of assembly functions such as GOTO and JUMP made it easier to understand the unstructured do loops in FORTRAN 77. I also found that the practice activities helped a lot while I learned Fortran. I will note that knowing other languages previously I felt Fortran 95 was much easier to learn and understand than FORTRAN 77 as it has more structure and better readability.