**CIS \* 3190 – Assignment 1 Reflection Report**

**Synopsis of Fortran**

My experience with Fortran was a very surprising and pleasant experience for me. From previous conversations I’ve had, I was of the great difficulty of reading and writing Fortran from Uncles and Aunts who took intro computer programming courses many years ago. I found that the Fortran 95 language was not only easy to understand, but simple to write in and modern. It was, in my opinion, just as easy as C if not easier. Although many of the features in Fortran that we used, such as intent, subroutines vs functions and external variables are not required in C, I found the inclusion of them made my code more understandable throughout development of my program. I believe that every programmer should take the time to learn the basics of Fortran as it is easy to learn and extremely prevalent in the computer science world (15% of all code).

**My Approach & Modifications**

Before starting this assignment, I decided to go through the overview of the Fortran language posted in Unit I, II and III on course link. This resource allowed me to learn about Fortran basics from syntax, to data types, subroutines, functions and much more. After completing these readings, I began the assignment by coding the front screen print statement that welcomed the player to the game. This allowed me to have something easy that I could test my gfortran compiler with. After installing the compiler and testing it with this short program, I had my environment set up to start my programming.

The first major modification I made to the specifications of the assignment is when I changed the 3x3 character matrix variable that represented the board into a 9-character array. I did this because I believed that it was more logical to use this approach, seeing as I could use the users input as well as the random number generation as the index of the array, instead of having to map their number to a position in the matrix that is unrelated. Reflecting on my decision I would have done the same thing because I found that I was able to write cleaner, more readable code without the 9 if statements that the sample could for chkPlay has.

The second major modification I made to the code that was provided to us was changing it so there were no GOTO statements and the syntax was the correct for the f95 version of fortran. A few of these changes included adding the “::” operator for variables, as well as changing the do loops, and adding intent in and out to subroutines.

Additional to the 7 functions/subroutines that were described in the specs of the assignment, I added two of my own that made it easier for me to modularize my code. These two were “initBoard()” and “getValue()”. InitBoard() is a subroutine that initializes the character array that represents to board to all spaces at the beginning of the game. GetValue() is a function that returns an integer for the amount the character is worth (4 for an O, 1 for an X and 0 for a space). This is used in the AI pickMove() function to help the computer decide what the best move possible is.

**What were your biggest problems?**

My biggest problem in my opinion was the approach I took the AI function. My approach was a very lengthy, brute force algorithm that attempted to find the best move the computer could would make. This could have been improved by using a more modular approach to the algorithm. For example, when I found a row with a block or a win I could have made a function that returns the location of the empty spot instead of having to look for it every time I search a different combination of three. Another problem I faced was seeding the random number generator that I used when there were no block or winning moves. The solution I was using worked on my computer but when I tested it on the SOCS servers it did not work.

**What were some good features?**

One of the features I enjoyed to use during my experience programming in Fortran included the “intent(in), intent(out), intent(inout)” declarations. I found this feature to be helpful as I was treat variables as if they were pointers without all the referencing and dereferencing. It also helped me when I organized my variable declarations within each subroutine or function into areas for local, parameters and function variables. Another feature that helped me was the ability to write subroutines or functions. The distinction between the using a function and subroutine helped me consider the most effective way to write my code and although it isn’t necessary in other languages, I found it interesting to learn and implement. Lastly, to my surprise Fortran had many of the same loops and syntax as C which I found very easy to use and readable. For example, if then loops, do while loops, and similar function declarations amongst others.

**Would it have been easier to use C?**

I believe that C would have been easier to use simply because I have more experience with it and not because it is a better or easier programming language. Disregarding my previous knowledge in C, I believe that Fortran was easier because of two main reasons. Instead of pointers that are used in C, there was a straight forward intent feature that made it much easier to handle variables that required changes in functions and subroutines. On top of this, in my experience with C, I found that I often ran into segmentation faults issues because of pointers and memory allocation, whereas I never did using Fortran.

**Was Fortran easy to learn?**

Fortran was very easy to learn in my opinion. I believe I had an easy time learning it because of the many similarities to C regarding syntax and logic.