Reflection Report for Assignment 2

By Ian McKechnie (1051662)

Writing in Ada turned out to be a lot harder than I thought it was going to be. I read all the Ada readings right after assignment 1 and started writing this assignment a week out from the due date. Little did I know, I forgot most of what I read so a lot of the code I learned I learned on the fly. Working with the data types in this, especially strings and characters, turned out to be a lot harder than I thought it was going to be. I/O with strings was hard at first and took me 2 days to fully understand what I was doing.

The approach I took for assignment I think was good, especially for a first time writing in the language. Because of all the gotos in the original program I just used if-else statements for every possible branch that the original program had. The idea was that I would then get it running and debugged, then take out repeating code blocks with procedures. However, by the time it was running it was almost 600 lines. This made fixing bugs very hard because there was so much repeating code that that an error could be in 4-5 sections of almost identical code. So, finding bugs took a lot longer than I was expecting. Once I finally finished the debugging, I was able to shrink the almost 600 lines by roughly 50%. Then once the program was shrunk and still working, I changed all the variable names from a single letter to a work that better represents the data the variable is storing. I think the approach took a lot more time that necessary. I should have instead just identified any repeating code blocks as I was writing and put them into procedures so then debugging would have been easier. I don't think Ada was well suited for this program. This is a program that revolves around strings and characters (guesses). Ada must have strings at a defined length and if strings are to small you must fill the rest with spaces. This means you must also store the lengths of the words in another array that will tell the program how long the strings are. This is a lot of wasted memory. I think a language that has dynamic string lengths such as C or Python would have been best for rewriting this assuming I had choice of any language.

The strengths of Ada in this project was it is strict and strong typed which allows Ada to give very precise error and warning messages. I prefer a strict and strong typed language when I am learning as it forces me to keep track of variable types and I can easily figure out a variable type is down the road as I just look at where that variable was declared. I also liked the precise error messages. This makes learning a language much easier as I can find the precise line and know exactly what I did wrong. I did like the procedures as well. It reminds me of passing variables by pointer in C. This allows me to take large blocks of repeating code out that change multiple variables and shrink it down. This made cleaning up my code very easy.

Overall, I would say I don't like Ada, I don't mind programming in it but it's not my favourite legacy language.