Yizhou Wang
1013411
wang15@uoguelph.ca

Overall, I don’t think Ada is well suited for this problem. I feel Ada has a very limited ability to handle the string and character. The more important thing is Ada puts a lot of attention on variable types and doesn't have lots of options to cast it by itself, which brings lots of trouble to get the program run correctly. For example, when I tried to create a string array in Ada, I could only make all the strings with the same length else I need to use unbounded string instead. However, Ada also won’t allow you to assign an unbounded string to a string easily(The only way to do it is first to make this string with a certain length, and then you can assign unbounded string to the string). Therefore, I end up switching all my string variables into unbounded strings.

Besides the string, another hard part of this assignment is to restructure the program. The original program uses lots of GOTO statements. It’s very hard to turn into a loop statement since there are lots of options to either continue the loop or not. Eventually, I made this program into one triple while loop. The most outer loop is to control the whole game, users can only leave if they choose to quit. The middle-level loop is to control each round of the game, this loop only ends when either user wins or loses the round. The most inner loop is for user input, this loop only leaves when the user enters new and correct characters.

In General, Ada is a great language. The inconveniences in variable typecasting make Ada more reliable. What’s more, Ada also supports run-time checks to prevent access errors, off-by-one errors, buffer overflow errors. That’s why Ada is widely used in critical systems. Ada is also an object-oriented programming language and has modular programming mechanisms. These features make Ada can be used to develop large software systems. I feel Ada is harder than Fortran. Also, it takes me a while to get used to using := as an assigned symbol instead of =.