Reflection Report – ADA Programming

Although ADA is a new programming language that I was not familiar with, I was able to adapt to this language and its syntax with ease and also believe ADA was well suited for solving basic image processing. There are several structures and syntax ADA uses that make it a beneficial choice for the problems presented in this assignment. I found some of ADA's syntax to be similar to the C programming language which is valuable as most of my programming knowledge is related to C. An example of this is between records in ADA and structs in C. In ADA, objects in a record are accessed using the period character, similar to C where a period is used to access variables in a struct. Additionally, making changes to an array is similar to C and effortless to implement. This allowed modifying image records, calculating the probability density function, and calculating the cumulative histogram to be completed with ease. Moreover, my favourite feature of ADA is its efficient use of error handling. In many other programming languages, it is difficult to avoid program failure due to multiple errors such as a datatype mismatch. I found ADA's use of exceptions simple to implement, as I used these to prevent name errors and data errors, allowing to make my program as unbreakable as possible. This includes retrieving user input and avoiding type mismatches, as well as preventing users from attempting to open a file that does not exist. I also appreciate ADA's detailed error messages displayed by the compiler. There are a variety of overflow checks which assisted me in fixing issues while working on this assignment. Presented in the terminal is the program name where the error was found, the line and column number, as well as a descriptive message related to the error and often suggesting how it can be fixed. Error messages that assisted my progress were those related to syntax, type mismatches and missing operands as working with multiple packages at once was overwhelming. Furthermore, I found it extremely easy to read in files and store data in ADA. In the case of this assignment, ADA's *get* function allows for efficiency as my program reads in each value from the file and stores the data into a record.

Aside from these benefits, there were a few limitations I came across. Firstly, the syntax for initializing variables was a continuous error I had to fix as usually I use only '=' instead of ':='. Another minor syntax error was my use of curly brackets for conditional statements instead of using the keyword *end*. Additionally, when beginning this assignment, it was challenging to share data between all packages and successfully compile my program. Extra time was required as well as research to discover how packages can share subprograms and type declarations. Although this was a struggle near the beginning of the assignment, I was able to overcome this issue and continue to work on this smoothly.

Ultimately, learning a new programming language may require more time to adapt however, my experience with ADA was positive. I enjoyed working with PGM files and seeing my program output results through 8-bit images along with multiple new concepts made this assignment a practical experience. In conclusion, ADA's efficient use of error handling and easy to understand programming structure will lead me to continue to utilize this legacy programming knowledge in future assignments.