Assignment 2: Ada Image Processing Drew Mainprize

1045298

For assignment 2, I created a simple image processing program that took PGM greyscale images and did various transformations to them. The transformations included inversion, log-shifting and image stretching. These transformations were performed by pulling in the image file into a record in Ada, implementing the required algorithm on the matrix of pixels contained in the file, and then outputting the resulting pixel matrix to a new PGM file with the transformation. I used procedures, functions, packages, records, multidimensional arrays and subtypes to implement the assignment.

Some of the issues I had while writing the program were using the libraries (specifically unbounded strings) and typecasting. Unbounded strings were fairly different from what is typical of a C program and required more massaging of inputs and outputs to get the results I wanted. For example, any time I needed to do any conversions or put/get I needed to use a to_string() or unbounded_to_string() method to convert the datatype. This was difficult for me to get the hang of as it seems like Ada handles strings quite well aside from this.

Typecasting was another issue I had. Typecasting had a bit of an overlap with unbounded strings but my main issues with typecasting was dealing with multiple input types from either the command line or a file. I discovered that the libraries for text_IO for the various variable types did not mix well together. I ran into issues trying to read in integers and unbound strings in the same procedure and found that the easiest solution was to have 'use' statements for some of them and manually use ada.datatype.text_IO when I needed to use a different type of input variable. Another solution was simply to read in everything I could as an unbound string and then typecast it to and integer or whatever I needed. This took a lot of extra time to implement and was one of the features of Ada that I did not like.

However, there were a few features that I did enjoy about the language. Firstly, I did find it much more readable that Fortran and found it easy to compare some of the components of Ada to C-based programming languages. Some of those being functions in C that act the same as procedures and functions as well as loops. I enjoyed that the file structure was very easy to understand. Particularly, .ads files resemble C header files and .adb files resemble a general .c file, linking everything was also very familiar. A new feature that I often don't take advantage of in other languages was being able to declare a data type in a .ads file and link it to every file so that they can all share the datatype.